A SECURE GROUP COMMUNICATION ARCHITECTURE FOR A SWARM OF AUTONOMOUS UNMANNED AERIAL VEHICLES

THESIS

Adrian N. Phillips, Captain, USAF

AFIT/GCE/ENG/08-09

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCE/ENG/08-09

**A SECURE GROUP COMMUNICATION ARCHITECTURE FOR A SWARM OF
AUTONOMOUS UNMANNED AERIAL VEHICLES**

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Engineering

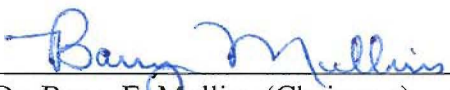Adrian N. Phillips, BS ECE, MBA

Captain, USAF

March 2008

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GCE/ENG/08-09

A SECURE GROUP COMMUNICATION ARCHITECTURE FOR A SWARM OF AUTONOMOUS UNMANNED AERIAL VEHICLES

Adrian N. Phillips, BS ECE, MBA

Captain, USAF

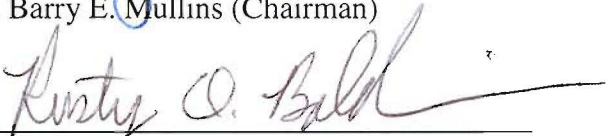Approved:

_____
Dr. Barry E. Mullins (Chairman)

3 Mar 08
Date

_____
Dr. Rusty O. Baldwin (Member)

3 Mar 08
Date

_____
Dr. Richard A. Raines (Member)

3 Mar 08
Date

Abstract

This thesis investigates the application of a secure group communication architecture to a swarm of autonomous unmanned aerial vehicles (UAVs). A multicast secure group communication architecture for the low earth orbit (LEO) satellite environment is evaluated to determine if it can be effectively adapted to a swarm of UAVs and provide secure, scalable, and efficient communications.

The performance of the proposed security architecture is evaluated with two other commonly used architectures using a discrete event computer simulation developed using MatLab. Performance is evaluated in terms of the scalability and efficiency of the group key distribution and management scheme when the swarm size, swarm mobility, multicast group join and departure rates are varied. The metrics include the total keys distributed over the simulation period, the average number of times an individual UAV must rekey, the average bandwidth used to rekey the swarm, and the average percentage of battery consumed by a UAV to rekey over the simulation period.

The proposed security architecture can successfully be applied to a swarm of autonomous UAVs using current technology. The proposed architecture is more efficient and scalable than the other tested and commonly-used architectures. Over all the tested configurations, the proposed architecture distributes 55.2 – 94.8% fewer keys, rekeys 59.0 - 94.9% less often per UAV, uses 55.2 - 87.9% less bandwidth to rekey, and reduces the battery consumption by 16.9 – 85.4%.

Acknowledgments

Table of Contents

List of Figures

List of Tables

A SECURE GROUP COMMUNICATION ARCHITECTURE FOR A SWARM OF AUTONOMOUS UNMANNED AERIAL VEHICLES

I. Introduction

1.1 Motivation

A swarm of autonomous unmanned aerial vehicles (UAVs) has great potential to provide benefits in a variety of applications, especially in the Department of Defense (DoD) intelligence, surveillance, and reconnaissance (ISR) mission. UAV swarm applications include continuous border patrol, battlespace surveillance, mapping routes for troop movement, real-time information distribution to mobile military units, and extending communications via an airborne network. Grouping UAVs into a swarm allows them to carry a range of sensors with an array of capabilities, creating a diverse group that can overcome the limited field of view of a single UAV [KeJ06]. A swarm also increases reliability through redundancy. Recently, UAVs have made significant contributions to the Global War on Terrorism and demand for them is only expected to increase, especially as swarming technology develops and matures [OSD05].

Previous UAV swarm research improved communication efficiency and effectiveness, with little emphasis on security [HyM07, YaB06, KeJ06]. However, the sensitivity of UAV swarm applications necessitates a secure communication architecture that provides DoD-mandated information assurance. With this added security component, a swarm of autonomous UAVs can provide a unique and powerful net-centric asset to support the warfighter.

1.2 Overview and Goals

Securing communication in a UAV swarm is of the utmost importance. Without it the swarm is vulnerable to malicious acts such as traffic analysis, denial of service attacks, or masquerading by adversaries. In the worst-case, these acts could lead to human casualties. With that said, the cost of securing a UAV swarm, given its limited resources, needs to be efficient and scalable so the mission can still be accomplished. Secure communications is not as complex for only a few UAVs. However, providing secure communication for a large, dynamic swarm of UAVs is significantly more complex.

This thesis focuses on a significant aspect of securing group communication in an autonomous UAV swarm; providing an efficient and scalable architecture for group rekeying needed for encrypting and decrypting group communication. This research expands on the Hubenko architecture [HuR07], which develops a multicast secure group communication architecture for low earth orbit (LEO) satellite networks in the global information grid (GIG). This architecture can be effectively adapted to a swarm of UAVs to provide secure, scalable, and efficient communications.

The goals of this research are to investigate the feasibility of using the Hubenko architecture to provide a secure, scalable, and efficient multicast architecture for UAV swarms, and to evaluate the security performance of the Hubenko architecture applied to a swarm of UAVs compared to two other commonly used security architectures.

1.3 Thesis Layout

This chapter introduces the research topic and the motivation behind the effort. In Chapter 2, background information and fundamental concepts are presented as well as

recent work in the topic area. Chapter 3 outlines the methodology used to carry out the experiments. Chapter 4 provides discussion and analysis of the experimental results. Chapter 5 draws conclusions about the results and offers areas for future research.

II. Background and Literature Review

2.1 Introduction

This chapter presents fundamental concepts and recent research in the areas of UAVs, Mobile Ad Hoc Networks (MANETs), multicasting technology, secure and scalable multicast architectures, and the GIG. Section 2.2 defines UAVs and discusses autonomy, UAV swarms, and their possible applications. Section 2.3 introduces MANETs and discusses their relationship to UAV swarms. Section 2.4 presents multicasting technology, multicast groups, multicast routing protocols, multicast security, and group key management. Select multicast architectures are discussed including an overview of the Hubenko Architecture. Section 2.5, provides an overview of the GIG and describes how satellite and UAV communications can fit into this model.

2.2. Unmanned Aerial Vehicles

2.2.1 Overview

UAVs have been around for decades and have been used in a variety of roles in both military and commercial applications. The DoD defines a UAV as:

> A powered, aerial vehicle that does not carry a human operator, uses aerodynamic forces to provide vehicle lift, can fly autonomously or be piloted remotely, can be expendable or recoverable, and can carry a lethal or nonlethal payload. [DoD01]

Recently, UAVs have made significant contributions in the Global War on Terrorism, including support in Operation Enduring Freedom and Operation Iraqi Freedom. These diverse systems, which cost a few thousand dollars to tens of millions of dollars, are capable of performing a wide range of missions without endangering the military's most valuable asset--its people [OSD05].

UAVs in the military are typically used for ISR missions, although they have also provided substantial support to intelligence preparation of the battlefield (IPB), situation development, battle management (BM), battle damage assessment (BDA), and rear area security (RAS) [Glo07]. Combat UAVs, known as UCAVs, have the additional capability to deliver weapons and are useful for high-risk missions.

Along with the array of roles, UAVs come in many different shapes and sizes with varying capabilities. The Air Force's Global Hawk is the largest operational UAV with a wingspan of 35.3 meters and an approximate weight of 11,600 kg [USA05b]. This is in contrast to some of the smallest UAVs, which are insect-sized "mesicopters", and miniature "smart dust" sensors [DeU05]. These tiny systems are broadly designated as Miniature Aerial Vehicles (MAV) measuring less than 15 cm in any dimension. Between these two extremes is a class of small UAVs (SUAV), which typically weigh under 50 kg and are powered by a battery or liquid fuel engine. Tables 1 and 2 outline UAVs and SUAVs respectively currently employed by the DoD along with their characteristics [OSD05].

Table 1. Selected UAVs and their Characteristics [OSD05]

UAV	Weight (kg)	Wingspan (meters)	Payload Capacity (kg)	Radius (km)	Endurance (hrs)	Ceiling (meters)
Global Hawk (RQ-4A)	12,160	35.4	886	10,000	32	20,000
Predator	4,773	20.1	341	3,704	16 - 30	15,240
Hunter (RQ-5A)	736	8.9	91	266.7	11.6	4,572

Table 2. Selected Small UAVs and their Characteristics [OSD05]

UAV	Weight (kg)	Wingspan (meters)	Payload Capacity (kg)	Engine Type	Radius (meters)	Endurance (minutes)	Power (Watts)	Ceiling (meters)
Desert Hawk III	2.7	1.37	1.0	Battery	10,000	90+	120 – 300	305
Pointer	3.8	2.74	0.45	Battery	11,000	120	145 – 300	305
Raven	1.8	1.32	0.9	Battery	11,000	90	80 – 200	305
Dragon Eye	2	1.17	0.45	Battery	4,600	45 - 60	110 - 200	305

The capabilities of SUAVs and MAVs currently in operation are limited by their size, unlike larger UAVs and manned aircraft. SUAVs have a communication range of about 10 km for the data link, which is dependent on battery power. The video link is line of sight (LOS) and is usually the first link to be lost. These links typically range from 1.6 km – 11 km. The smaller the UAV the less battery power, processing power, data storage capacity, and data link capability it will have, thereby restricting the communication range and data processing capabilities.

Despite the fact that SUAVs and MAVs have limited capabilities, the technology is advancing. Many of the above limitations are improving and UAVs of all sizes and types are becoming much more attractive and feasible. Operational commanders and senior military officials recognize the full worth of UAVs and their contributions to the missions. This is due in part to several recent technological advances such as [OSD04]:

- Dramatic increases in computer processing power

- Battery and other energy storage systems are becoming more efficient, cheaper, and lighter in weight

- Sensor technology advancement including reduced sensor size and weight, increased resolution, and the detection of fixed and moving targets under harsh environmental conditions

- Improved communications, image processing, and image exploitation capabilities

In additional to these technological advances, there are other factors that are making UAVs vital future platforms [OSD04]:

- Political and public pressure to minimize casualties and capture of aircrews

- Emerging requirement for continuous persistent surveillance of the battlespace

- Their proven worth in the Global War on Terrorism

Another area where UAVs have significant potential is when they are used in autonomous swarms. The application of swarm theory to UAVs has generated significant interest, especially among military researchers. The potential for groups of UAVs to sense and respond autonomously, without human intervention, is very powerful. Autonomous UAVs and UAV swarms are discussed in the next two sections.

2.2.2 Autonomous UAVs

Autonomy is commonly defined as the quality or state of being self-governing or having the ability to make decisions without human intervention [Mer07]. UAVs have varying degrees of autonomy. When UAVs were first developed they were often referred to as drones, because they were simply remote controlled aircraft, with no autonomy whatsoever. UAVs considered to be "fully autonomous" have been developed, such as the Army's RQ-11 Raven. However, the use of the term fully autonomous herein means being able to take-off, fly, and land without human intervention.

What actually constitutes full autonomy is not yet agreed upon. For example, several federal agencies including the Department of Defense Joint Program Office, the Air Force Research Laboratory, the U.S. Army Science Board, the Army Maneuver Support Center, and National Institute of Standards and Technology all have their own definitions and degrees of autonomy for various programs. This has given rise to the creation of the Federal Agencies Ad Hoc Autonomy Levels for Unmanned Systems (ALFUS) Working Group [Nat04]. ALFUS defines autonomy as:

> The unmanned system's own ability of sensing, perceiving, analyzing,
> communicating, planning, decision-making, and acting, to achieve its
> goals as assigned by its human operator(s) through designed human-robot

interaction. Autonomy is characterized into levels by factors including mission complexity, environmental difficulty, and level of HRI to accomplish the missions. [Nat04]

The directors of the Service Research Laboratories have defined their own levels of autonomy for unmanned aircraft (UA) in the Unmanned Aircraft Systems Roadmap [OSD05]. Figure 1 shows these 10 levels along with examples of where UAVs currently in operation fall.
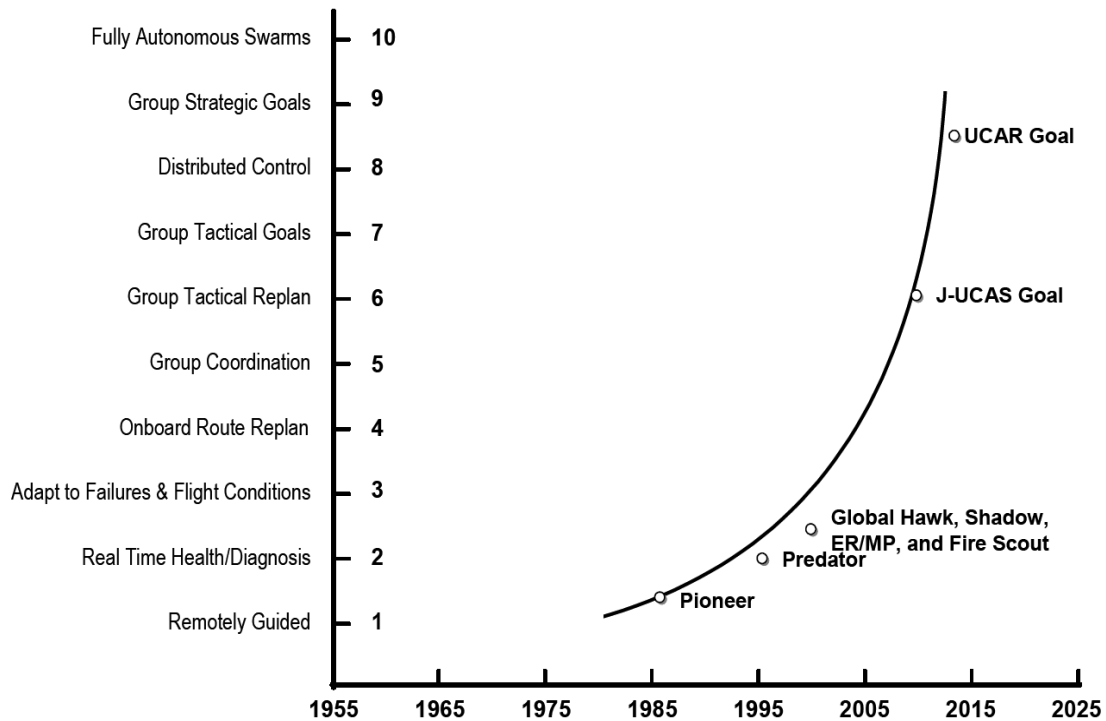
Figure 1. Autonomy Levels of UAVs as defined by the OSD [OSD05]

As can be seen from the figure, all of the higher levels of autonomy are characterized by collaborative group behavior and fully autonomous swarm technology is expected in the next decade. The autonomous UAV swarm concept is discussed in the next section.

2.2.3 UAV Swarms

With UAV research and development evolving at a rapid pace, an autonomous swarm of UAVs for military applications is receiving much interest. Swarm is usually

identified with a group of living organisms who arrange themselves to cooperate to achieve a common task that could not be completed as an individual [KeJ06]. In the context of this research, a UAV swarm consists of a heterogeneous group of autonomous UAVs working together (cooperatively) to accomplish a common task or mission.

A swarm of autonomous UAVs has great potential in a variety of applications. The area that is getting the most attention and research is ISR. Applications of UAV swarms include, but are not limited to: continuous border patrol, battlespace surveillance, mapping routes for troop movement, real-time information distribution to mobile military units, and extending communications by providing an airborne network. Grouping UAVs into a swarm allows them to carry a range of sensors, with an array of capabilities, creating a diverse group that can overcome the limited field of view of a single SUAV [KeJ06]. The swarm also provides increased reliability through redundancy.

A Host of Armed Reconnaissance Vehicles Enabling Surveillance and Targeting (HARVEST) [AuM06] is theoretically capable of autonomous refueling, cooperative search, information fusion, and munitions employment. HARVEST consists of a heterogeneous group of small to medium size UAVs equipped with various payloads, some with munitions and some with sensors for data collection. The core cooperative functions include localization, area search, and data routing with potential applications to include target tracking, active decoys, tactical networking, terrain mapping, and environmental hazard plume detection [AuM06]. A swarm of UAVs such as HARVEST is a specific type of mobile ad hoc network (MANET). Therefore, MANETs are presented in more detail in the next section.

2.3 Mobile Ad Hoc Networks

A MANET is system of mobile hosts connected by wireless links, the union of which forms a communication network modeled in the form of an arbitrary communication graph [DeG03]. MANETs are of great interest to the military because there are many situations where there is no fixed infrastructure network and it is not feasible to build or maintain.

MANETs are complicated systems because they lack any fixed infrastructure, in contrast with more traditional wireless networks where mobile nodes communicate via fixed access points or base stations. The mobility of nodes in a MANET causes the network topology to change often and can be very unpredictable. As the mobility of the nodes increases, the protocol overhead necessary to support the changing network increases as well. This is important since most nodes in a MANET rely on batteries, and are therefore limited by power. The limited power necessitates an efficient communication method. One way to achieve efficient communications in a MANET is through multicast communication.

2.4 Multicasting

Multicasting is a set of technologies that allow a source node to send data to multiple destination nodes simultaneously while transmitting only a single copy of the data on to the network [SuH03]. The data is replicated for the destination nodes only when necessary. While multicasting can be applied to different layers of the OSI stack, the term typically refers to multicasting at the network/IP layer (IP Multicasting) [Wik07a]. Multicasting is ideal for group communications because it allows a source node to efficiently send data to a large group of users.

Multicasting is often grouped with unicasting, broadcasting, and anycasting as a kind of network layer communications. Unicasting transmits data from a single source node to a single destination node. This type of communication is appropriate for applications such as downloading content from a web site. However, for group applications, unicasting can be very inefficient because the source needs to transmit a separate message for every member of the group. Broadcasting transmits data from a single source node to every node on the local network. This can also be inefficient because it sends data to every node on the network even if the intended group is only a small subset. This results in a significant amount of wasted bandwidth. Anycasting transmits data from a single source node to one or more of a group of destination nodes, usually the closest. While anycasting has its place, it is not suitable for group communications, where all the members of the group need to receive the data.

In a MANET where efficiency is a very important factor due to limited resources such as power and bandwidth, multicasting can be very beneficial since multicasting efficiently uses network bandwidth and reduces processing load on the source.

2.4.1 Multicast Communication Process and Requirements

Implementing multicast communications on a network can be a very complex process. There are several requirements and components of this process that must exist to successfully establish multicast communications.

The first requirement for multicasting is the existence of multicast supported routers within the network to route packets. With multicasting still in its infancy, most Internet routers do not support IP multicast. However, this will soon change with the adoption of IPv6, in which IP multicast will be a standard feature [Wik07b]. In the mean

time, the multicast backbone (M-bone) supports IP multicasting over the Internet. M-bone is a virtual network laid on the existing backbones that operate the Internet and intranets [WoD07]. It uses a subset of the class D address range (224.0.0.0 to 239.255.255.255) for multicast traffic.

Another requirement for multicast is the identification of the receivers, or the group. Hosts that would like to join a multicast group must identify themselves to the network. This is called the registration process, and it is facilitated with the unique set of IP addresses (mentioned above) that are reserved specifically for multicast communications [Spi03]. This process is managed by a special group management protocol, which will be described in a later section.

After the receivers join their respective groups, the network must deliver multicast traffic using a multicast routing protocol, which is the final component to the multicasting process. The routing protocol determines the appropriate forwarding paths to all members of the multicast group. The protocol also determines when it is necessary to replicate data, so that the information can be received in multiple locations simultaneously [Spi03]. Group management protocols and multicast routing protocols are described in more detail in the following sections.

2.4.2 Multicast Groups

The purpose of multicasting is to transmit data to a group of users, thus the routers must know who constitutes the multicast group. Membership in a multicast group must be dynamic, allowing hosts to enter and leave the multicast session without the permission or knowledge of other hosts [Kru98]. Hosts must also be allowed to belong to more than one group at a time. These key features of multicasting are managed by a

group membership protocol. The most popular protocol is the Internet Group Management Protocol (IGMP).

IGMP supports three main types of messages: Report, Query and Leave. An IGMP Report message is issued by a host wishing to join a multicast group and includes the IP multicast class D address of the multicast group. An IGMP Query message is issued by the multicast router to hosts on its network to determine whether the hosts still wish to receive multicast traffic. An IGMP Leave message is issued by a host when it no longer wishes to receive multicast traffic for that group.

Multicast Listener Discovery (MLD) is another multicast group management protocol used in the IPv6 protocol suite. MLD is similar to IGMP although it uses different message types. Until IPv6 is widely implemented, IGMP will be the predominate group management protocol for multicast.

2.4.3 Multicast Routing Protocols

A multicast routing protocol is an intelligent control panel mechanism that efficiently delivers data from the first hop router (FHR) to all of the participating last hop routers (LHR) [Spi03]. The FHR is the router closest to the source of the multicast traffic, while the LHRs are the last routers in the path to the receivers. The best multicast protocol ultimately depends on the particular application. There is no single multicast protocol that outperforms all others in all applications. Each protocol has its own strengths and weaknesses.

When considering which multicast protocol to implement, the following properties are desirable for most applications: low cost with respect to processing power; low end-to-end delay; scalability; ability to support dynamic group membership;

survivability in terms of network, link, or node outages; and some level of fairness to all members [SaM00]. The next section presents an overview of the most popular multicast protocols in use today followed by multicast protocols best suited for a MANET.

2.4.3.1 Multicast Protocols suited for MANETs

There are several well established protocols which provide efficient multicasting in the fixed, wired environment, such as Distance Vector Multicast Routing Protocol (DVMRP), Multicast Open Shortest Path First (MOSPF), Protocol Independent Multicast - Sparse Mode (PIM-SM), and Protocol Independent Multicast - Dense Mode (PIM-SM) [WaP98, MoP94, FeH06, AdN05]. However, adapting these to a mobile ad hoc environment is challenging and not always feasible. Although existing multicast support for fixed users can be applied to a wireless environment, there are many issues that make this challenging including limited bandwidth, higher rate of packet loss, higher rate of membership changes, changes in routing structure, unpredictable topology, and limited battery life [Var02].

Most multicast routing protocols for MANETs fall into one of the following categories: tree-based, meshed-based, stateless multicast, or a hybrid of these. Many of the protocols used in fixed, wired environments are tree-based and stateful protocols because the routers can maintain the distribution trees. Tree-based multicast protocols for MANETs are similar. In this type of protocol, a source-based or shared tree is used among the group and only one path exists between any pair of nodes. Examples of tree-based ad hoc multicast protocols include Ad Hoc Multicast Routing Protocol Utilizing Increasing ID Numbers (AMRIS) [WuT98] and Multicast Ad Hoc On-Demand Distance Vector (MAODV) [RoP00].

14

Mesh-based protocols establish a mesh-like structure, which contains multiple, redundant routes between nodes for robust handling of link failures or node mobility. Examples of mesh-based ad hoc multicast protocols include Core-Assisted Mesh Protocol (CAMP) [MaG99] and On-Demand Multicast Routing Protocol (ODMRP) [YiL02].

Stateless multicast protocols avoid the overhead of creating and maintaining the delivery tree, or mesh, by explicitly including the list of destinations in the packet header. These protocols are tailored for small group multicast and assume the underlying routing protocol takes care of forwarding the packet to respective destinations based on the addresses contained in the header [DeG03]. A pure flooding protocol is an example of a stateless multicast protocol along with more optimized stateless protocols such as Differential Destination Multicast (DDM) [JiC01].

Hybrid protocols attempt to combine the robustness of meshed-based protocols with the efficiency of tree-based protocols. Examples of hybrid protocols are Ad Hoc Multicast Routing Protocol (AMRoute) [BoM98] and Multicast Core Extraction Distributed Ad Hoc Routing (MCEDAR) [SiS99].

De Morais Cordeiro, Gossain, Agrawal [DeG03] compared several protocols proposed for multicasting in mobile ad hoc networks based on the following factors: type of topology, possibility of loop formations, dependence on a unicast protocol, whether control packets are flooded throughout the network or limited to the multicast group, and whether paths are created on demand, or if optimal paths are determined once and updated periodically as needed. In a harsh environment, where the network topology changes very frequently, mesh-based protocols outperformed tree-based protocols due to the alternate paths, which allowed multicast data to be delivered to all or most multicast

receivers even if some links failed. Hybrid protocols were suitable for medium mobility networks by taking advantage of both a tree and a mesh structure. Stateless multicast is promising for supporting multiple small multicast groups [DeG03].

2.4.4 Multicast Security

The security of multicast communication can be as important, if not more important, than performance and efficiency. This is especially true in the military. Before discussing the specifics of securing multicast communication in a MANET, some basic security concepts warrant further review.

Whether discussing wired versus wireless, unicast versus multicast, or infrastructure versus ad hoc, there are basic security services that should be built into all communication architectures to protect the information. The three basic components of computer security are [Bis03]:

- *Availability*: the ability to use the information or resource desired,

- *Confidentiality*: the concealment of information or resources, and

- *Integrity*: the trustworthiness of data or resources.

The NSA has defined five pillars of Information Assurance (IA), which includes confidentiality, integrity, and availability, plus:

- *Non-repudiation*: a service that provides proof of the integrity and origin of information, and

- *Authentication*: the ability to verify the identity of the user, device or other entity.

In addition to the security services defined above, there are security services that are unique to a multicast environment. These services include:

- *Group Key Secrecy*: the guarantee that it is computationally infeasible for an adversary to discover any group encryption key [AyS06],

- *Forward Secrecy*: new members are not able to read past traffic [BrR02],

- *Backward Secrecy*: former members are not able to read present and future traffic [BrR02], and

- *Group Access Control*: ability to permit or deny membership into multicast groups [JuA02].

For secure wireless multicasting, cryptography and key management schemes are needed, in which cryptographic keys are used to encrypt and decrypt messages. Public key cryptography is a secure mechanism used exchange encrypted messages, but is computationally expensive. In a multicast protocol when the amount of data to be transmitted is large, or when the devices involved in communication cannot perform computationally-intense exponentiations, symmetric key cryptography can secure communications [LaP06]. This will be the case in UAV swarms until public key cryptology is no longer a significant burden on a UAV's limited resources.

Symmetric key cryptography requires all group members to use the same decryption key. This shared decryption key is called the Session Encryption Key (SEK) or Traffic Encryption Key (TEK). Since everyone shares the SEK, members need to hold additional Key Encryption Keys (KEK) that are used to securely distribute the SEK to each valid member [LaP06]. The key management scheme manages and distributes keys in addition to ensuring only legitimate members of the multicast group hold valid keys and can access the group data during a multicast session.

The dynamic nature of MANETs increases the complexity and overhead of managing this process. To preserve the secrecy of the multicast data, the SEK needs to be updated upon certain events such as a member joining and leaving the group. For secure multicasting in a wireless environment, other factors heavily impact the ideal key management scheme such as: battery power, bandwidth, constraints, host mobility, loss of packets, and wireless security issues [AyS06].

Encryption and digital signatures provide adequate confidentiality, integrity, and authentication. Because these security mechanisms have proven to be very effective when implemented properly, the majority of the literature focuses security concerns on protecting the key material [Kru98]. The protection, management, and distribution of the key material is discussed in the next section.

2.4.5 Group Key Management

Group key management is a significant component of secure multicast communication and has a large impact on scalability. The distribution and management of the cryptographic keys is one of the most challenging issues in multicast security. There has been significant research in this area, and like routing protocols, the best key management scheme depends upon the application.

A group key management scheme defines the key agreement mechanism at the beginning of a multicast session. Additionally, it defines the successive key exchanges during a session when the group changes without rebuilding the group [BrR02]. This operation is usually defined as the rekeying operation and completely characterizes a key management protocol [BrR02]. Group key management schemes can be categorized into three main groups [RaH03]:

- *Centralized*: A single entity controls the whole group, hence the group key management protocol seeks to minimize storage requirements, computational power on both clients and servers, and bandwidth utilization.

- *Decentralized*: The management of a large group is divided among subgroup managers, to minimize the problem of concentrating the work in a single place.

- *Distributed*: There is no explicit key distribution center, and the members generate keys. All members perform access control and the generation of the key can be either contributory, meaning that all members contribute some information to generate the group key, or keys can be generated by one of the member of the group.

Although there has been much research in the area of group key management, there has been little on schemes which provide secure multicast communications in mobile ad hoc networks. The next section discusses multicast schemes which have the potential to contribute to a new secure and scalable multicast architecture suitable for a MANET.

2.4.6 Select Secure, Scalable Multicast Architectures

A variety of solutions have been proposed to provide a secure and scalable multicast architecture. Each has its strengths and weaknesses, and applications in which it can be best applied. The approaches most applicable to this research are reviewed in the following sections.

2.4.6.1 GOTHIC

As mentioned above, group access control is one of the necessary services for secure multicast. Under ordinary IGMP, any host wishing to gain access to a multicast

group and its data can do so by submitting a JOIN request. Thus, eavesdropping, theft of service, and denial of service (DOS) attacks can occur. Any viable solution to this problem must provide a *group policy management system* and a *group member authorization system*. The group policy management system has a group owner who provides a list of authorized members and other appropriate security policies for the group to the access control server (ACS) [JuA03]. The group member authorization system provides the core control of the architecture by controlling access to the group [JuA03].

GOTHIC is a comprehensive architecture that provides group access control [JuA03]. It contains a group policy management system and a group member authorization system. The GOTHIC group policy management system occurs first. A host wishing to be a group owner submits a request to the ACS. The ACS authenticates and authorizes the group owner via the group owner determination and authorization system (GODAS). The GODAS ensures the host attempting to provide the group policy is really the group owner. Implementation depends on the particular multicast allocation address scheme. Once the GODAS ensures the host is actually the group owner, the group policy is submitted. The ACS may be a single system or distributed across several systems. After the group policy is in place, the group member authorization system controls access to the multicast data. More specifically, a host wishing to join the group submits an authorization request to the ACS signed with the host's private key, which contains the group ID and the host's public key certificate [JuA02]. The ACS checks the group policy to determine whether the host has access rights to join the group. If it does, it is authenticated and given an expiration time.

Another key feature of GOTHIC is the group access control aware group key management (GACA-GKM). GACA-GKM leverages the trust built into the group access control system to reduce the requirements of the group key management scheme and obtain substantial overhead reductions [JuA03]. GACA-GKM has three basic rules [JuA02]:

1. If a host h joins multicast session G from a trusted subtree that has previously been part of the multicast tree for session G, a rekey must occur.

2. If a host h leaves multicast session G from a trusted subtree that will remain part of the multicast tree for session G, then a rekey must occur.

3. Otherwise, there is no need to rekey.

For example, if a new member, host A, is on a shared broadcast link with current group member, host B, then a rekey must occur when A joins since A had access to the distribution tree before it became a member. Also, if a leaving member, host C, is on a shared link with current member host D, then a rekey must occur when C leaves otherwise it will have access to the distribution tree after it is no longer a member. These three rules include users in the same trusted subtree in addition to users on a shared broadcast link due to the possibility of eavesdropping in the form of wiretaps and network sniffing [JuA02]. This feature saves a tremendous amount of rekeying overhead and therefore, can significantly increase the scalability of the secure multicast architecture. The designers of Gothic created the architecture with low computation overhead at the routers, low message overhead, and low support infrastructure requirements [JuA02]. These attributes are ideal for resource-constrained MANETs and, in particular, UAV swarms.

2.4.6.2 Spatial Clustering

Another recent multicast architecture that shows great promise is called "Spatial Clustering" [BaB02]. While GOTHIC provides group access control, spatial clustering focuses on group key management. The Spatial Clustering architecture reduces the overhead involved when group members join or leave the group by dividing the multicast group into subgroups based on their physical location. These subgroups are independent of each other and have their own group leader and their own secret group key. Each subgroup is managed by a group security agent (GSA), which work together with other GSAs to bridge the local multicast traffic from each subgroup into all of the other subgroups as needed [Hub06]. At the head of the entire hierarchy is a group security controller that is responsible for managing all of the GSAs and the overall security of the group.

Under the normal flat multicast architecture where there are no subgroups, the entire multicast group must rekey when a join or leave occurs. Every rekey operation incurs $O(N)$ key distributions, where N is number of members in the group. It is easy to see that this flat architecture does not scale well to large, dynamic groups. Thus, by creating a hierarchical architecture, as Spatial Clustering does, each independent group need rekey only when there is a change in its own subgroup.

Other schemes, such as Iolus, use a similar subgrouping concept, but the subgroups are based on predetermined administrative boundaries [Mit97]. Spatial Clustering dynamically forms the subgroups and assigns group leaders. Because these assignments are base on location, the key distribution scheme can exploit the parallelism inherent in different parts of a multicast tree and greatly enhance performance [BaB02].

2.4.6.3 Hubenko Multicast Security Architecture for GIG Environment

The Hubenko architecture is a secure group communication architecture that combines the key features of the well known multicast architectures in a way that increases system scalability for secure multicast in a LEO satellite environment [HuR07]. One of its key features is clustering. The best features of Spatial Clustering, discussed in Section 2.4.6.2, and Iolus [Mit97] are combined to form the basic framework of the Hubenko architecture. Multicast groups are divided into subgroups (clusters) based on the physical location of its members. By using spatial boundaries the key distribution scheme can exploit the parallelism inherent in different parts of a multicast tree to greatly enhance performance [BaB02]. Using the Iolus framework, all of the clusters are independent and each cluster has its own group leader and SEK. As a result, if a new member joins or leaves the multicast group, only the affected cluster needs to rekey as opposed to the entire multicast group. Each cluster is managed by a GSA, known as a cluster leader in this research. The cluster leaders work with other cluster leaders to bridge the local multicast traffic from each cluster into all of the other clusters as needed [Mit97]. At the head of the entire hierarchy is a group security controller that manages all of the cluster leaders and the overall security of the group. This is the job of the satellites in the Hubenko architecture. The number and size of the clusters as well as the number of levels in the hierarchy is flexible depending on the application.

To further increase system scalability, the Hubenko architecture incorporates many of the key features of Gothic, discussed in Section 2.4.3.1. The GACs from Gothic are used to strengthen the security of the system by preventing unauthorized users from attempting malicious acts such as traffic analysis or denial of service attacks. GACA-

GKM is also incorporated, which significantly improves scalability and efficiency, through less frequent rekeying by taking advantage of the trust built into the GAC system [HuR07]. For example, in a typical group key management system, whenever a user joins or leaves a multicast group, the entire system is rekeyed since the new user could have accessed either the old encrypted data prior to arrival or to new encrypted data after departure. GAC ensures no unwanted users have access to the data prior to their validated join or after their departure. Thus, a rekey is not needed in either of these situations [HuR07].

Figure 2 shows a conceptual model of the Hubenko architecture in a LEO satellite environment.
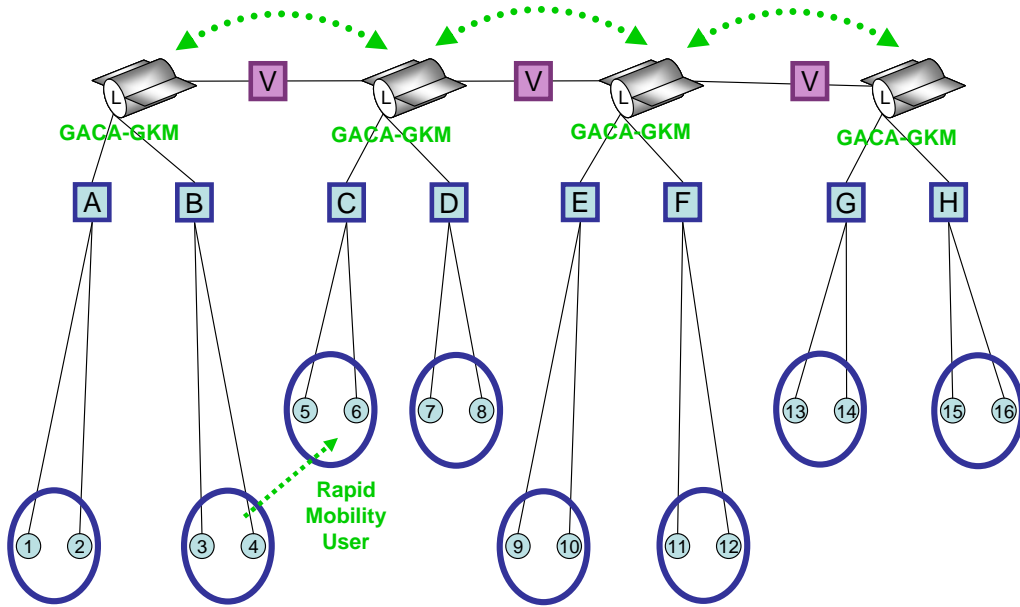
Figure 2. Hubenko Architecture [HuR07]

The LEO satellites are represented by satellite figures, the group keys are represented by the letters, and the users are represented by the numbers. In this architecture, LEO satellites form a cluster at the top of the hierarchy with a group key "V". The users are

further divided into clusters, with a different group key for each cluster. The Hubenko architecture is a modular design. As a result, the underlying multicast routing protocol and rekeying protocol are transparent and can be selected to best fit each unique application. Although this architecture has been applied to a LEO satellite system, due to its modularity it can also be scaled to provide a secure, scalable multicast architecture for UAV swarms in the GIG.

2.5 Global Information Grid

The DoD is currently in a great transformation that will reorient the military and focus its attention on emerging and future missions, change the way they operate and fight to leverage Information Age concepts and technologies, and change business processes to create an Information Age organization [AlH03]. To achieve this transformation soldiers, platforms, weapons, sensors, computers, and communications systems must be connected to share strategic, operational, and tactical information in real-time. To make this transformation a reality, the DoD is in the process of building the infrastructure for this net-centric environment known as the GIG. The DoD defines the GIG as:

> …the globally interconnected, end-to-end set of information capabilities, associated processes and personnel for collecting, processing, storing, disseminating, and managing information on demand to warfighters, policy makers, and support personnel. The Global Information Grid includes owned and leased communications and computing systems and services, software (including applications), data, security services, other associated services and National Security Systems. [DoD01]

This network is increasingly becoming one of the most important contributors to combat power and protection. The GIG provides the communications backbone to carry out net-centric operations and enables users to exploit the tenets of the net-centric warfare

doctrine. Conceptually, the GIG will be similar to the Internet. However, there will be

less dependence on ground-based, fixed systems and equipment to transmit and route

data, and more dependence on space-based and mobile, ad hoc systems to carry out these

functions [GAO04]. Figure 3 is a conceptual view of the GIG including nominal assets.
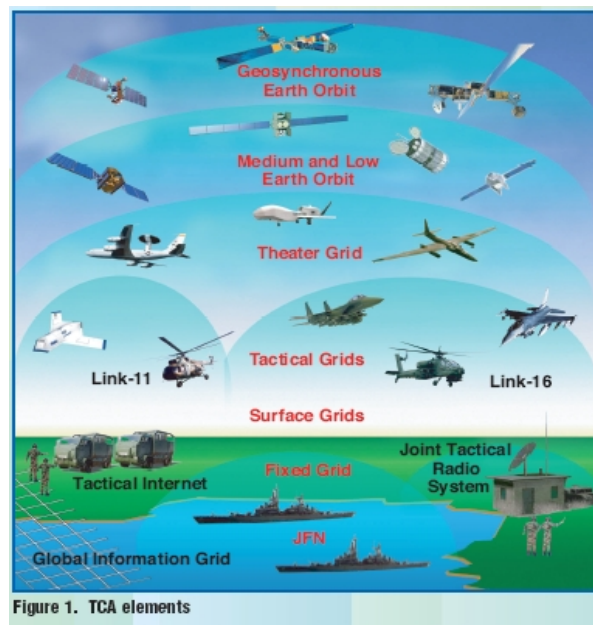
Figure 3. Global Information Grid Assets [RoM05]

There are several layers that make up the GIG's infrastructure. Hubenko [HuR06b]

defines and decomposes the GIG into four communications infrastructure "Layers":

- *Surface Layer*: contains wired and wireless communication assets for both
  tactical and strategic functions

- *Aerospace Layer*: contains tactical communication assets typically functioning
  within a theater of operation

- *Near-Space Layer*: contains the high loiter assets such as UAVs, primarily
  functioning as reconnaissance assets

- *Satellite Layer*: hosts orbiting satellite assets and provides a robust, high speed

  backbone in space

Hubenko [HuR06b] focused on enhancing the satellite layer by taking advantage of LEO

satellites, which provide greater communication efficiency compared to other satellite

systems. These systems are discussed in the following section.

2.5.1 Satellite Communications in the Global Information Grid

Satellites are a well established, mature technology, and will be a major part of

the GIG's infrastructure. These nodes enable real-time information transfer to and from

the warfighters and decision-makers while removing geographic constraints associated

with fixed terrestrial infrastructures [HuR06b]. Satellites are typically differentiated

based on the size and orbital path (circular or elliptical). The three main types of circular

orbit satellite systems are Geosynchronous Earth Orbit (GEO), LEO, and Medium earth

orbit (MEO). GEO satellites provide narrowband, wideband, and protected

communication capabilities [HuR06b]. These satellites orbit at an altitude of

approximately 35,000 km. Only three are needed to provide near full earth coverage.

However, coverage around the polar regions (beyond 20 degrees north and south latitude)

is degraded.

LEO systems operate at a much lower altitude than GEOs. LEO satellites

typically orbit at an altitude of about 1000 km. Unlike GEO satellites, which are fixed

relative to the Earth, LEO satellites travel across the Earth and only provide coverage of a

particular spot for a few minutes. Thus, 40+ systems are needed to provide full earth

coverage, including the polar regions. A hand off must occur between satellites if a

transmission between the satellite and user takes longer than a few minutes. However,

because of the lower operational altitude, the propagation times are greatly reduced. Aside from shorter propagation delay, LEO systems also minimize the possibility of a lost transmission because they provide coverage of every spot on Earth by at least two satellites at any given time. In addition, smaller ground equipment is required for end users making these systems ideal for mobile users.

MEO, another category of circular orbit satellites, falls between GEO and LEO systems. MEO satellites typically orbit at an altitude of 10,000 km and provide a view of a particular spot for several hours. While, fewer satellites are required to provide full earth coverage compared to LEO systems, the higher altitude increases the propagation delay. More detailed descriptions of these systems can be found in [HuR06b].

GEO systems, in particular, are most commonly associated with the GIG compared to LEO and MEO systems. However, Hubenko [HuR06b] proposes enhancing the satellite layer of the GIG by incorporating more LEO systems for the following reasons:

- GEO can provide *near*-real-time performance, while LEO systems can deliver *real*-time performance for voice and video transmissions.

- LEO systems can achieve average end-to-end latencies of less than 100 ms for intercontinental communications using satellite crosslinks.

- Real-time voice and video communications for tactical users in the field is an increasing requirement, which LEO satellites are better able to provide.

- LEO systems can provide "Power to the Edge."[1]

---

[1] "Power to the edge" empowers individuals, or edge devices, at the edge of an organization (where the organization interacts with its operating environment to have an impact or effect on that environment) [AlH03]

In addition, LEO satellites can provide a gateway for other assets in the GIG such as UAV swarms. The closer proximity coupled with the smaller communication equipment may allow SUAVs to reach out to a LEO satellite to connect to the GIG when a ground station is not available or is a less efficient means of communicating.

2.5.2 UAV Communications in the Global Information Grid

With the GIG emerging as the future platform to conduct all information operations, UAV communication must be seamlessly integrated to exploit all of their capabilities. UAVs are employed in a variety of roles, where information is the central focus, whether the information is being sent, received, acquired, or processed. In almost all circumstances, this information is sensitive and needs to be protected, making information assurance (IA) a crucial requirement of UAV communication. While, UAVs such as the Global Hawk are large enough to support the required storage capacity, processing power, and data links to provide secure long-range communications, SUAVs are much more constrained. This makes a secure, scalable, and efficient communication architecture crucial to the incorporation of SUAVs and UAV swarms into the GIG. The research done in the area of satellite communications in the GIG [HuR07] can scale to or extend to UAVs and may provide the key to efficiently securing their communication.

2.6 Summary

This chapter presents the fundamental concepts and recent research in the areas of UAVs, MANETs, multicasting technology, multicast architectures, and the GIG. UAV related topics such as autonomy, swarms, and MANETs are discussed. Next, multicasting technology is introduced along with group management, multicast routing

protocols, multicast security, and group key distribution. Secure and scalable multicast

architectures that can provide secure and efficient communication in a UAV swarm are

studied. Finally, the GIG and description of how satellites and UAVs can fit into this

model is also presented.

III. Methodology

3.1 Introduction

This chapter presents the methodology used to evaluate the security performance of three different security architectures applied to a swarm of autonomous UAVs. First, the problem definition, goals and hypothesis, and approach are discussed in Section 3.2. Section 3.3 defines the system boundaries. The system and its services are described in Section 3.4 followed by a detailed description of the workload in Section 3.5, the performance metrics in Section 3.6, the parameters in Section 3.7, and the factors in Section 3.8. Then, the evaluation technique is discussed in Section 3.9 followed by a description of the simulation environment in Section 3.10. Section 3.11 provides an overview of the experimental design. Finally, the technique used to analyze and interpret the data is covered in Section 3.12.

3.2 Problem Definition

3.2.1 Goals and Hypothesis

For UAV swarms to be powerful net-centric assets, they must be capable of secure, efficient, and scalable communication. Chapter 2 discusses several challenges to providing secure, efficient, and scalable communication in UAV swarms, highlighting the complexity of the problem. The security of the UAV swarm must be the top priority. Without it the swarm is vulnerable to malicious acts such as traffic analysis, denial of service attacks, or masquerading by adversaries, which in the worst-case could lead to human casualties. The cost of securing the UAV swarm, in terms of its limited resources, needs to be efficient and scalable so the swarm can still carry out its mission effectively.

Securing communication among only a few UAVs is not very complex. However, providing secure communication for a large, dynamic swarm of UAVs is significantly more complex.

This thesis focuses on a significant aspect of securing group communication in an autonomous UAV swarm; providing an efficient and scalable architecture for group rekeying. This research expands on previous work by Hubenko [HuR07], which developed the Hubenko architecture discussed in Section 2.4.6.3.

The goals of this research are to:

- Investigate the feasibility of using the Hubenko architecture to provide a secure, scalable multicast architecture for UAV swarms in the GIG, and

- Evaluate the security performance of the Hubenko architecture in a swarm of UAVs compared to two other commonly used security architectures.

It is hypothesized that the Hubenko architecture can be effectively adapted to a swarm of autonomous UAVs, while providing significant performance improvements in terms of group key management compared to two other commonly used security architectures. It is also hypothesized that the Hubenko architecture provides the most performance improvements in large, highly mobile swarms with high rates of joins, departures, and rejoins.

3.2.2 Approach

The Hubenko architecture combines key features of well known multicast architectures in a way that increases system scalability for secure multicast. Although this architecture has been applied to a LEO satellite system, this research investigates the

feasibility of using the architecture to provide a secure, scalable multicast architecture, which constitutes the approach for this research.

The Hubenko architecture applied to a UAV swarm is shown in Figure 4.
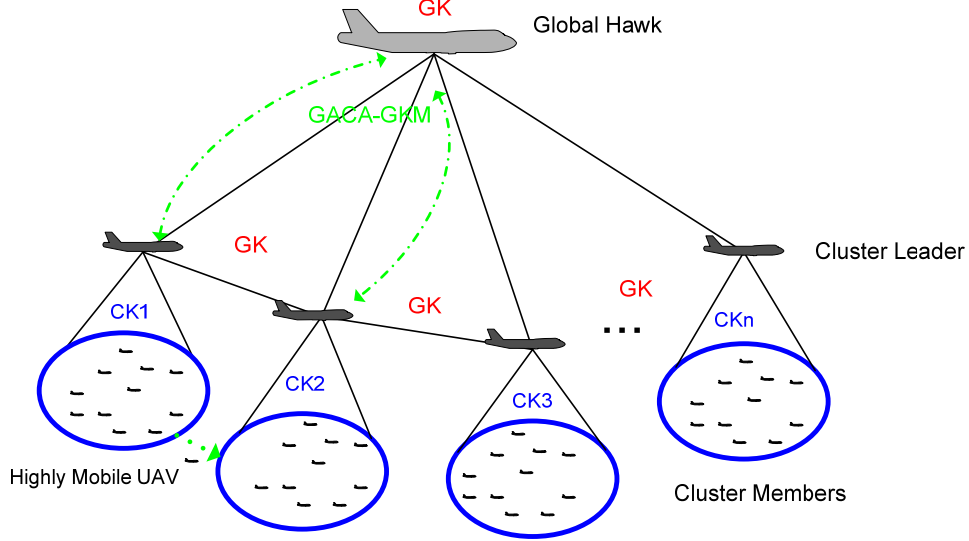
Figure 4. Hubenko Architecture Applied to UAV Swarm

A large UAV, such as a Global Hawk, has a similar role as the LEO satellite does in the original Hubenko architecture. The Global Hawk is the group security controller and group ACS, responsible for the overall security of the entire swarm. "GK" represents the multicast group key shared among the cluster leaders and the Global Hawk. Clusters are formed based on spatial boundaries to maximize communication efficiency. Each cluster has its own cluster key represented by "CKn". The black lines represent communication links while the circles with thick lines represent cluster boundaries. The dashed lines represent the implementation of GACA-GKM on the Global Hawk, which communicates with the cluster leaders to manage access to the group. Instead of satellite spot beams dictating the number and size of the clusters, cluster boundaries are constrained by the capabilities of the UAVs selected as cluster leaders. That is, the radio transmission range

is dependent on numerous factors including transmission power, receiver sensitivity, and antenna design. These factors will vary from UAV to UAV.

When a multicast group first forms, the Global Hawk assigns UAVs as either cluster leaders or cluster members based on their capabilities and location. Ideally medium sized UAVs are assigned as cluster leaders, because they have greater range, endurance, and processing capabilities. The UAVs selected as cluster leaders communicate with the Global Hawk flying at an altitude of about 15 km and all of the UAVs in their respective clusters. To increase available bandwidth and avoid transmission collisions, the cluster leaders loiter above their clusters and use directional antennas aimed at their cluster. The cluster leaders communicate amongst each other to keep their clusters from overlapping.

Although large autonomous UAV swarms are still in the concept stage, it is important to ensure the technology exists to allow the proposed communication architecture to be applied. This research assumes the swarm consists of a secure Global Hawk, several medium-sized UAVs (such as the Hunter) as cluster leaders, and largely, SUAVs (such as the Desert Hawk) and MAVs. Based on the capabilities of these UAVs, displayed in Table 1 and Table 2, the Global Hawk operates at about 15 km, the medium-sized UAVs at about 4 km, and the SUAVs at about 300 meters. In the worst-case scenario, when medium-sized UAVs are not present to act as intermediary routers, a SUAV may need to be capable of a communication range of up to 15 km to communicate directly with the Global Hawk. Using current radio and battery technology, this communication range is feasible [Ubi07, Gru07]. This issue is further discussed in

Appendix B, which provides specific details on the representative battery and radio including their capabilities.

The other architectures evaluated in this study are the baseline and the cluster. The baseline architecture for a swarm of UAVs is a flat model, consisting of the swarm and the multicast group leader, which is the Global Hawk. It includes the basic security functions of key generation, key storage, key agreement, and group key distribution to provide a dynamic application proof-of-concept [HuR07]. The entire swarm shares a single SEK and thus every swarm member is rekeyed on a member join or departure.

The cluster architecture is an enhanced baseline architecture that includes the clustering concepts from Spatial Clustering and Iolus. Each cluster is independent and has its own unique SEK. As a result, each cluster only needs to be rekeyed when there is a join to, or departure from its cluster.

The work by Hubenko in [HuR07] provides insight into the impact of the multicast group size and mobility on each of the investigated architectures. However, the activity and characteristics of the multicast groups modeled in that work do not reflect a realistic scenario for a swarm of UAVs. Hubenko's study models a multicast group whose members join within a fixed time, with some of the members leaving after random intervals. This is visually represented by Scenario 1 in Figure 5. There may be some applications when this model properly characterizes a UAV swarm, but the multicast activity represented by Scenario 2 in Figure 5 is a better model of a UAV swarm's activity. Scenario 2 in Figure 5 represents a multicast group with continuous departures, and rejoins to the group. Most of the envisioned missions of UAV swarms (continuous border patrol, battlespace surveillance, ISR, etc.) require the swarm sustains itself for

prolonged periods of time. This could be several hours or even several days. Currently SUAVs which comprise the bulk of the swarm, have limited battery life typically ranging from 1 to 3 hours [OSD07]. Therefore, in order for the swarm to sustain its strength and size, its members will need to depart and rejoin several times throughout the duration of the mission to replace or recharge batteries.
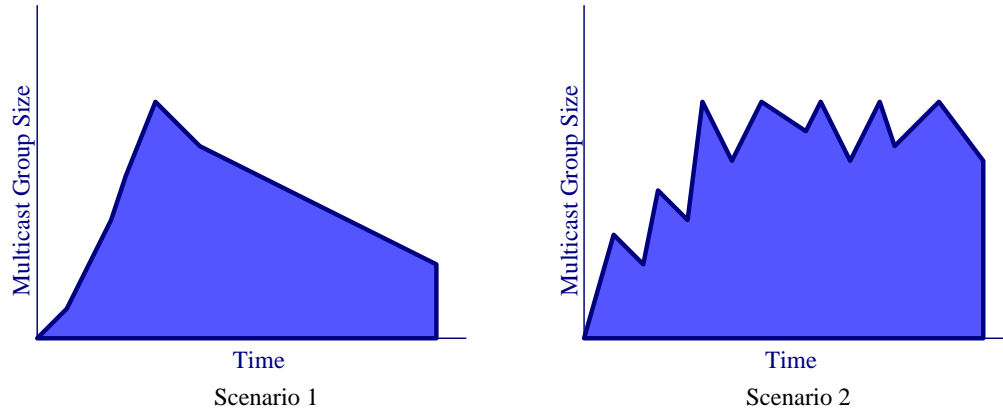
Figure 5. Tested Scenarios

Thus, this study tests the Hubenko architecture under Scenario 2 in addition to Scenario 1, to represent different mission requirements placed on UAV swarms. The scenarios are distinguished by the multicast group activity over the simulation period. Scenario 1 represents the scenario where UAVs join the swarm and must depart after their batteries are depleted. None of the departing UAVs rejoin the group. In Scenario 2, the UAV swarm joins the multicast group, but there are continuous departures and rejoins over a longer period. The burden of continuous departures and rejoins to the multicast group fully test the architectures for a UAV swarm.

3.3 System boundaries

The System Under Test (SUT) is the UAV Swarm Group Communication System. A block diagram of the SUT is shown in Figure 6. It consists of the following

components: the security architecture, wireless network, UAVs, and the multicast routing protocol. The component under test (CUT) is the security architecture. Specifically, the Hubenko architecture is compared to a baseline architecture and a clustered architecture.
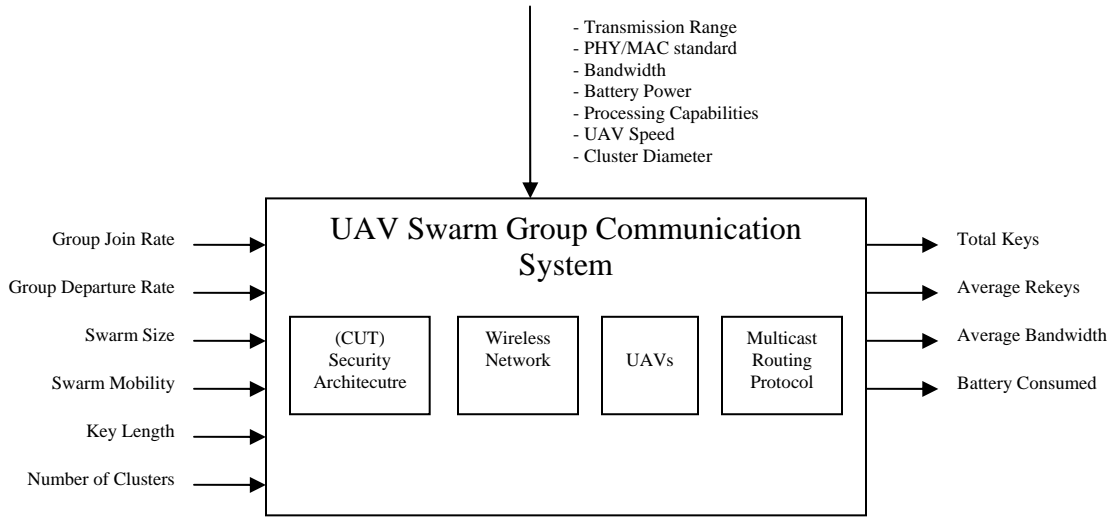
Figure 6. UAV Swarm Group Communication System

Workload parameters include the size of the swarm (multicast group), the group join rate, the group departure rate, the swarm's mobility, the number of clusters, and the length of the group key. The system parameters consist of the transmission range, bandwidth, the physical layer and MAC standard, battery power, processing capabilities, cluster diameter, and UAV speed. These parameters are discussed in more detail in Section 3.7. The metrics of the system consist of the total number of keys distributed over the simulation period (*total keys*), the average number of times a UAV must rekey (*average rekeys*), the average amount of bandwidth used to rekey (*average bandwidth*), and the average percentage of battery consumed to rekey (*battery consumed*).

In addition to the assumptions stated in Section 3.2, this study assumes there are no obstacles to UAV travel or communication and a reliable routing protocol is in place.

It is also assumed the UAVs are equipped with transceivers capable of successfully communicating over the necessary transmission range.

3.4 System Services

This system provides a secure multicast communication network service for a swarm of autonomous UAVs. This service is successful when encrypted multicast traffic reaches all intended nodes and only those nodes. For this to occur, the underlying security architecture must successfully distribute encryption keys to all group members and perform rekey operations as necessary, such as when a join or departure to the multicast group occurs. Based on the assumptions such as sufficient bandwidth and a reliable routing protocol, failures modes such as dropped packets are not considered.

3.5 Workload

The workload of the SUT is the amount of multicast traffic to be distributed. This study specifically focuses on reducing the traffic associated with group key management and distribution. Thus, the amount of multicast traffic related to group key management depends upon several parameters including the size of the swarm, the group join rate, the group departure rate, the swarm's mobility, the number of clusters, and the length of the group key. The workload to the SUT is generated by varying these parameters. For example, increasing the swarm size, the group join rate, the group departure rate, and the swarm's mobility increases the amount of rekey operations over the simulation period necessary to secure the swarm. As a result, the overall amount of multicast traffic increases, thus increasing the workload to the SUT. On the other hand, decreasing the swarm size, the group join rate, the group departure rate, and the swarm's mobility

38

decreases the amount of rekey operations over the simulation period, thus reducing multicast traffic and the workload to the SUT.

3.6 Performance Metrics

As group key management is one of the most complex and resource intensive operations on the network, the performance metrics should measure how efficient and scalable the security architecture is in terms of group key management. Thus, the following performance metrics are defined:

- *Total Keys*: The total number of keys distributed during the simulation period, and

- *Average Rekeys*: Average number of times a UAV must rekey during the simulation period.

Similar metrics were used to evaluate the performance of the Hubenko architecture in the LEO satellite environment as well as related work in the area of secure group communications [Hub06]. These metrics are also relevant to determining potential security performance improvements [HoI04, RaH03, Hub06]. In addition to the metrics listed above, Scenario 2 also measures:

- *Average Bandwidth*: The average amount of bandwidth used to rekey for a group rekey operation, and

- *Battery Consumed*: The average percentage of battery consumed by a UAV to rekey during the simulation period.

These metrics are very important in an environment such as an autonomous UAV swarm where battery capacity and bandwidth can be limited and costly, and further emphasize the cost associated with rekeying. The assumptions made, and the calculations used for

average bandwidth and *battery consumed* can be found in Appendix A and Appendix B, respectively.

3.7 Parameters

The parameters of the system are the properties, which when changed can impact the performance of the system. These include both system parameters, which characterize the system, and workload parameters, which characterize the workload. The system and workload parameters for the SUT are further described, with the fixed experimental parameters displayed in Table 3.

Table 3. Fixed parameter values

Parameter	Scenario 1 Value	Scenario 2 Value
PHY/MAC Standard	IEEE 802.11b	IEEE 802.11b
Bandwidth	11 Mbps	11 Mbps
Transmission Range	15 km	15 km
Processor Speed	1.8 GHz	1.8 GHz
UAV Speed	25 m/s	25 m/s
UAV Battery Capacity	4200 mA-hr	4200 mA-hr
Group Key Length	256 bits	256 bits
Number of Clusters	10	10
Cluster Diameter	10 km	10 km
Simulation Length	7200 time steps (2 hrs)	43200 time steps (12 hrs)

3.7.1 System Parameters

- Transmission Range: This is the maximum distance over which two nodes can successfully communicate directly. This is highly dependent on the UAV's battery, radio, and antenna. The representative radio is the SuperRange9, which features proven non-line-of sight distances over 20km [Ubi07]. This research assumes UAVs are capable of communicating up to distances of 15 km.

- Bandwidth: The channel bandwidth restricts how much data can be transmitted to the swarm per second. IEEE 802.11b has a maximum bandwidth of 11 Mbps.

- Battery Capacity: This affects the ability of the UAVs to generate and distribute keys and multicast data. UAVs used in similar research are currently equipped with a Thunder Power Lithium Poly battery TP4200-4S2PB, with a usable voltage range from 14 to 16.7 V [Gru07].

- PHY/MAC Standard: The physical layer and media access control standards define channel access and data encoding, modulation and transmission. IEEE 802.11b is a widely known technology and the current standard of choice for similar research [Pac07].

- Processor Speed: This also affects the ability of the UAVs to generate and distribute keys and multicast data. UAVs used in similar research are currently equipped with a Kontron 1.8 GHz processor with 1 GB memory [Gru07].

- UAV Speed: UAV speed impacts how fast and to what degree the network topology changes. A reasonable speed given the expected size and maneuverability of a typical UAV in HARVEST is 25 meters per second [AuM06].

- Cluster Diameter: This is a function of the antenna, transmission range, and altitude of the UAV chosen as the cluster leader and affects the swarm's coverage area. Based on these constraints the cluster diameter is chosen to be 10 km.

3.7.2 Workload Parameters

- Swarm Size: In this study the swarm size is synonymous with the multicast group size. The number of UAVs in the swarm can impact the cluster density, the total

number of keys which need to be distributed, in addition to the overall multicast traffic generated.

- Group Join Rate: The rate at which UAVs join the multicast group significantly impacts the overhead necessary to maintain overall security of the swarm.

- Group Departure Rate: The rate at which UAVs depart the multicast group significantly impacts the overhead necessary to maintain overall security of the swarm.

- Swarm Mobility: For the baseline architecture, mobility will not affect the need to rekey the group, because clusters do not exist and all UAVs are rekeyed by the same key server. However, mobility in the Cluster and Hubenko architectures significantly affects the number of rekeys, because each cluster is rekeyed separately. Thus, a highly-dynamic environment where UAVs are flying across several clusters, requires much more rekeying overhead than an environment where UAVs loiter in the same general area for long periods of time.

- Group Key Length: This affects the security of the system as well as the size of the packets generated to rekey the multicast group. Larger keys increase the security of the system, but require more bandwidth, processing power, and storage. This study assumes a key length of 256 bits, which is a standard length for AES encryption.

- Number of Clusters: This impacts the scalability, efficiency, and communication overhead required in the Cluster and Hubenko architectures. The ideal number of clusters varies depending on the situation and may be constrained by resources since each cluster is managed by a cluster leader, which requires more battery

power, processing power, storage, and endurance. Since cluster analysis is

beyond the scope of this research, the number of clusters for this study is set at 10

to allow for comparison to previous work [HuR07].

- <u>Simulation Length</u>: Longer simulations have more activity such as joins and

  departures and more mobility among the clusters. The simulation length for

  Scenario 1 is 2 hours which is the typical endurance of smaller UAVs. The

  simulation length for Scenario 2 is 12 hours. This represents UAVs having the

  ability to swap out batteries and rejoin the swarm after a certain amount of time.

3.8 Factors

This section outlines the factors selected from the system and workload

parameters. These factors are varied to determine the impact they have on the

performance of each security architecture evaluated. Table 4 and Table 5 summarize the

factors chosen and their levels for Scenario 1 and Scenario 2 respectively.

Table 4. Factor Levels Scenario 1

Factor	Level 1	Level 2	Level 3	Level 4
Swarm Size	40	100	200	500
Swarm Mobility	25%	75%		
Group Join Rate	15%	30%		
Group Departure Rate	25%	75%		
Security Architecture	Baseline	Clustered	Hubenko	

Table 5. Factor Levels Scenario 2

Factor	Level 1	Level 2	Level 3	Level 4	Level 5
Swarm Size	40	100	200	500	1000
Swarm Mobility	25%	50%	75%	90%	
Security Architecture	Baseline	Clustered	Hubenko		

- <u>Swarm size</u>: The number of UAVs in the swarm impacts the total number of keys

  to be distributed and also increases the overall activity of the swarm, thereby

43

increasing the number of times a UAV needs to rekey. Based on proposed UAV swarms and possible missions the levels selected are 40, 100, 200, and 500 UAVs. Scenario 2 also includes 1000 UAVs to further increase the workload.

- Swarm Mobility: This is the percentage of the swarm that is highly mobile. In this study, UAVs are defined as highly mobile if they travel outside of a 5 km radius, whereas UAVs that stay within a 5 km radius are defined as loiterers. A highly mobile environment requires much more rekeying overhead than one in which UAVs loiter in the same general area for long periods of time. The levels selected for Scenario 1 are 25% and 75%. In addition to these levels, Scenario 2 includes 50% and 90% swarm mobility levels.

- Group Join Rate: This is the percentage of the simulation time it takes for the entire swarm has initially joined the multicast group. The rate at which UAVs join the multicast group impacts the overhead necessary to maintain overall security of the swarm. The levels chosen are 15% and 30%. Thus, when the rate is set to 15%, there will be several more joins to the multicast group in a shorter amount of time compared to when the rate is sent to 30%. The group join rate for Scenario 2 is fixed.

- Group Departure Rate: This is the percentage of the swarm that departs the multicast group prior to the end of the simulation. The number of departures from the multicast group has an impact on the overhead necessary to maintain overall security of the swarm. The levels chosen for Scenario1 are 25% and 75%. The UAVs that depart the group do so after a normally distributed amount of time.

The group departure rate for Scenario 2 is not a factor because it is set to 100% for all of the simulations.

- Security Architecture: This is the CUT. The security architecture impacts the total number of rekeying operations and the overall security performance of the system. The levels selected are the baseline (flat architecture), cluster, and Hubenko.

3.9 Evaluation Technique

Currently a swarm of autonomous unmanned vehicles is still in the concept stage and an actual system is not yet fielded. Thus, measurement of an actual system is not feasible for this study. In addition, using an actual system, if one existed, would be very costly and time consuming. Using an analytical model is also not a viable option, because there is no such model that can be adapted to this scenario.

The best evaluation technique for this study is simulation. Both OPNET Modeler and MatLab tools have been considered for performing the simulation. Because this study is specifically concerned with reducing security overhead in the form of group key management, much of the details about data transmission, packets, and routing can be abstracted away. This makes MatLab the best choice to perform the simulation for this study.

A discrete event computer simulation using MatLab, (version R2007a) is developed to evaluate the relative performance of the baseline, cluster, and Hubenko architectures in terms of group key management and distribution in a swarm of UAVs. The characteristics of the UAV swarm are modeled in MatLab using structures which are initialized based on random variables and probability distributions appropriate for each

factor. Matrices track performance metrics which are saved as Excel files. The data from the Excel files are imported into Minitab where plots and figures are created for post-simulation analysis. These simulations can be reproduced on any workstation with MatLab version R2007a. The simulation environment is further described in Section 3.10.

The results of the simulation are validated by comparing them to related work in [Hub07]. Furthermore, using literature and other work done with group key management, certain scenarios can be validated. For example, increasing the number of joins and departures to a group should increase the number of rekeys.

3.10 Simulation Environment

The simulation environment developed for this research is a modified version of the one used in [Hub07], which modeled a satellite-based multicast network. Several modifications to the simulation were made to characterize a swarm of UAVs and this study's experimental design. Although a detailed description of the original simulation environment can be found in [Hub07], several significant modifications are described below.

In the original simulation the time steps were left undefined, however for the purpose of this research one time step represents one second. This means if a UAV joins the multicast group at the beginning of the one second interval it will not receive a multicast key until the end of the interval, thus having to wait up to one second to start receiving multicast data. The same logic applies to a UAV leaving the multicast group. If a UAV leaves the multicast group at the beginning of the one second interval it still may be able to receive multicast data for up to one second because the rest of the UAVs

in the multicast group will be rekeyed at the end of the one second interval. In actual use applications larger or smaller intervals can be used depending on the security needs of the system. The length of the rekey interval also effects the amount of traffic in the entire system. See Appendix C for a more detailed explanation.

Another important modification is how the metrics *total keys* and *average rekeys* are calculated for the baseline study. Because the original study dealt with a geographically widespread satellite environment, the baseline architecture required a rekey operation anytime a user moved from one spot beam to another whether or not the user was already a member of the multicast group. However, in this study the baseline architecture is a large UAV acting as the single multicast group leader with a swarm of smaller UAVs locally spread out within its range. Because it is assumed that the multicast group leader can directly and/or indirectly transmit to all members of the swarm, there is no need to rekey as swarm members move within that range. For example, the highly mobile UAV in Figure 4 would not cause a rekey in the baseline study because the clusters are non-existent and Global Hawk acts as the multicast group leader for the entire swarm.

The simulation environment is also modified to simulate both scenarios. The original study only simulated the multicast group activity of Scenario 1 shown in Figure 5. Aside from the changes mentioned above both scenarios required changes to the experimental parameters and factors to correspond to the experimental design.

3.10.1 Scenario 1 Simulation

In Scenario 1, two hours or 7200 discrete time steps of rekeying activity in a UAV swarm is simulated. During each simulation run all factors (join rate, departure rate,

swarm mobility, and swarm size) are held constant, but the three architectures are tested under the same conditions.   Each UAV is randomly assigned an initial join time to the multicast group, an initial cluster, a mobility type (highly mobile or loitering), and a departure time (if applicable). All of the random assignments are based on a uniform distribution.   The group join rate determines whether the UAVs randomly join within the first 15% or 30% of the simulation time. The group departure rate determines the percentage of the swarm that departs the group before the end of the simulation (either 25% or 75%). The swarm mobility rate determines the percentage of the swarm assigned as highly mobile or loiterers. The UAVs assigned as highly mobile change clusters throughout the simulation based on their velocity of 25 m/s, while the UAVs assigned as loiterers remain in their initial assigned cluster. The metrics *total keys* and *average keys* are tracked for each individual UAV for each of the three tested architectures.

3.10.2 Scenario 2 Simulation

In Scenario 2 12 hours or 43200 discrete time steps of rekeying activity in a UAV swarm is simulated. This scenario allows UAVs to rejoin the multicast group after departing and models the situation where a UAV swarm needs to be sustained for a long period of time, longer than a UAV's typical battery life. Thus, UAVs depart the swarm to recharge or exchange their batteries and then rejoin the group. The join rate is not a factor and is held constant (all UAVs join within the first hour). The departure rate is also not a factor in this scenario because all UAVs continuously depart and rejoin the multicast group, therefore making it 100%. Similar to Scenario 1, the swarm mobility and swarm size are held constant during each run and the three architectures are tested simultaneously under the same conditions.

In the beginning of the simulation each UAV is randomly assigned an initial join time during the first simulated hour (3600 time steps). Each UAV is also randomly assigned a duration (battery life) ranging from 30 minutes to 180 minutes to represent various battery capacities, typical of a heterogeneous swarm. This represents the battery capacities of the various SUAVs currently in operation as can be found in the DoD's Unmanned Systems Roadmap [OSD07]. Each UAV is randomly assigned to an initial cluster and as highly mobile or loitering. After a UAV initially joins the multicast group it stays for its randomly assigned duration and then departs. It then rejoins the swarm 30 minutes later representing the time to swap out its battery. This is repeated throughout the simulation. The metrics *total keys*, *average keys*, *average bandwidth*, and *battery consumed* are tracked for each individual UAV for each of the three tested architectures.

3.11 Experimental Design

The overall experimental design for this study consists of two sub-experiments, each with a full-factorial design with the factors stated above. The first sub-experiment, Scenario 1, consists of 8 repetitions for each configuration, requiring a total of 768 simulation runs (4 * 2 * 2 * 2 * 3 * 8 = 768). The second sub-experiment, Scenario 2, consists of 20 repetitions for each configuration, requiring a total of 1200 runs (5 * 4 * 3 * 20 = 1200). Thus, the overall experiment consists of 1968 simulation runs. The number of repetitions provides a narrow enough confidence interval while minimizing the number of experiments necessary. Each of the repetitions for the same configuration use a different seed for the random number generator which affects the various aspects of the simulation including the join time, departure time, and mobility of the each UAV.

3.12 Analysis and Interpretation of Results

The analysis of the data supports the goals of this research. Several random variables dictate the behavior of each individual UAV throughout the simulation. These include the UAV's join time, assigned cluster, mobility, departure time, and rejoin time. Thus, the data collected are random variables. Each experiment is replicated several times, as indicated above, to achieve an accurate representation of the system's typical performance. Errors in the sampled data are verified to be normally distributed and confidence intervals are used to compare the performance of the three architectures. If the confidence intervals for two architectures do not overlap they are said to have a significant statistical difference for the performance metric. If the confidence intervals overlap, but the means are not within the part that overlaps a t-test is performed to determine if a statistical difference exists. Otherwise, the architectures cannot be deemed statistically different and thus, one architecture cannot be said to perform better or worse than the other in terms of the measured performance metric.

To allocate the variation in *total keys*, *average rekeys*, *average bandwidth*, and *battery consumed*, an Analysis of Variance (ANOVA) is performed on each metric. This shows if the variance in performance is due to experimental error or real differences in the changing factors. In order for the results of the ANOVA to be valid, several assumptions must hold. The assumptions of the ANOVA are: the errors are randomly, independently, and normally distributed with a mean of zero, and have a common variance. These assumptions can be verified by examining various data plots including a normal probability plot, a residual versus fits plot, a histogram of the residuals, and a residuals versus order of the data plot.

3.13 Summary

      This chapter discusses the methodology used to evaluate the performance of secure group communication architectures applied to a swarm of autonomous unmanned aerial vehicles UAVs. Performance is evaluated via simulations using MatLab and is based on the multicast security performance metrics: *total keys*, *average rekeys*, *average bandwidth*, and *battery consumed*. A full-factorial experiment is performed on two different scenarios to evaluate the impact of varying the swarm size, group join rate, group departure rate, swarm mobility, and the selected security architecture.
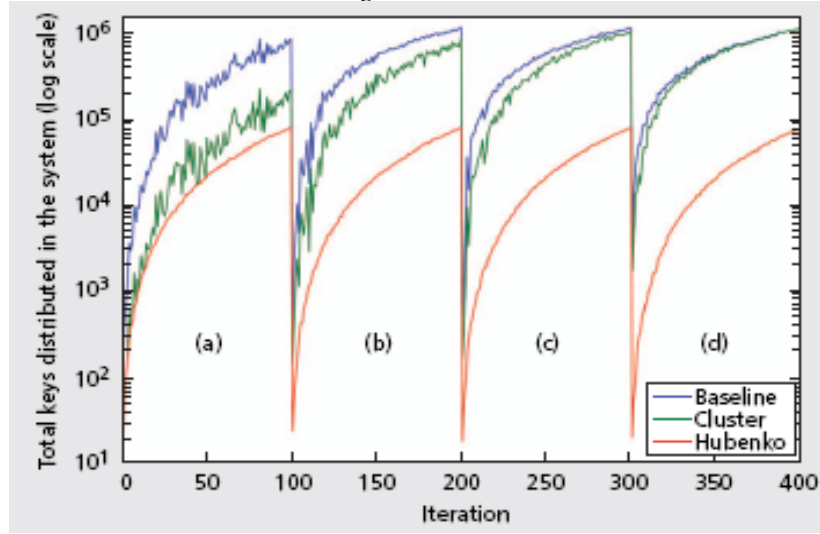
IV. Results and Analysis

4.1 Introduction

This chapter presents and analyzes the experimental results. First, the methods used to validate the architecture models are discussed in Section 4.2. Next, the results of each individual performance metric for Scenario 1 and Scenario 2 are presented in Section 4.3. Finally, an overall analysis of the results is provided in Section 4.4.

4.2 Architecture Model Validation

The purpose of this section is to validate the models of the architectures evaluated in this study. This is accomplished by duplicating the experiments documented in [HuR07] and comparing the results to the original findings.

The original experiments modeled stationary, ground, sea, and air users, each with different types of mobility (speed of the users) and rates of mobility (percentage of users that are mobile). The three architectures (baseline, cluster, and Hubenko) are tested in 400 different simulated scenarios, increasing the number of users in the system across the four different rates of mobility (1%, 10%, 25%, and 75%). By comparing the two plots in Figure 7, it is seen that the results from the duplicated experiments closely match the results from the original experiment for the metric *total keys*. The original results are displayed in the top plot and the duplicated experimental results are displayed below. The labels (a) – (d) correspond to the mobility levels, 1%, 10%, 25%, and 75%. Each iteration within each the mobility category increases the number of users with the first iteration at 100 users and the last iteration at 1000 users.
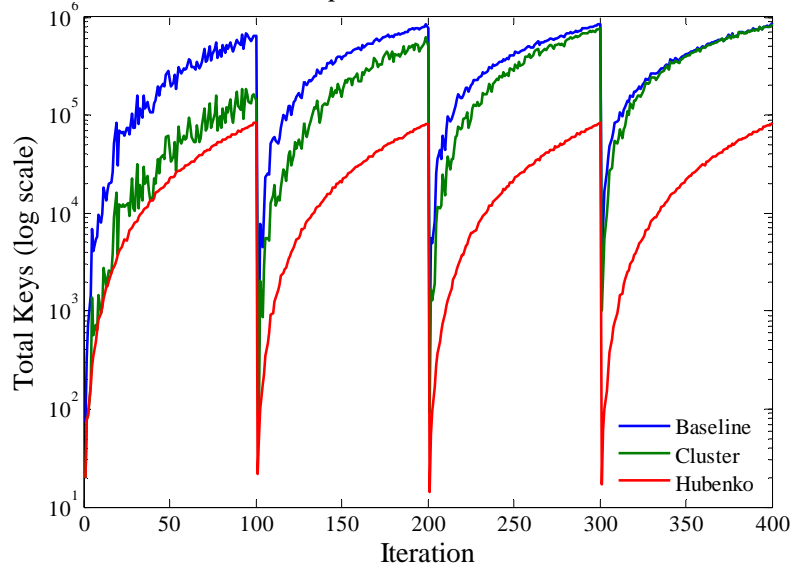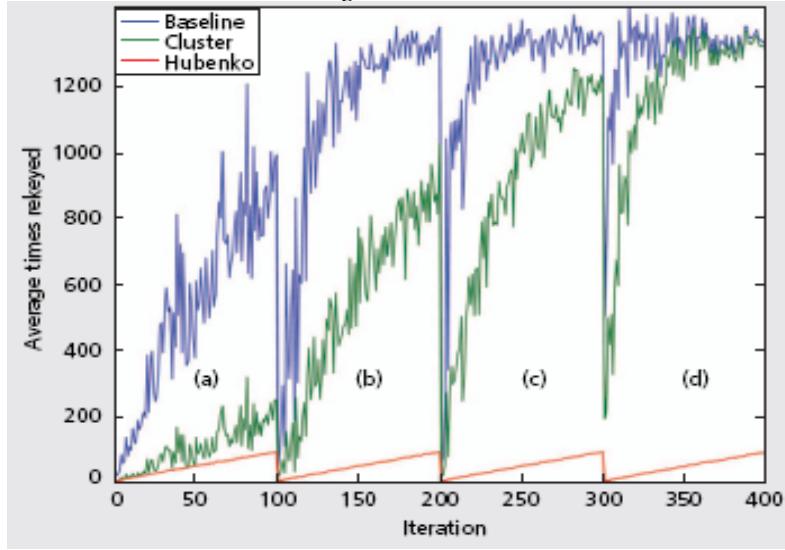
Figure 7. Validation of Architecture Models via Log Total Keys

Similarly, Figure 8 displays the results in terms of the metric *average rekeys*.

Although *average rekeys* were fewer overall in the duplicated experiment, the trends and

relationships between the three architectures closely match the original results.
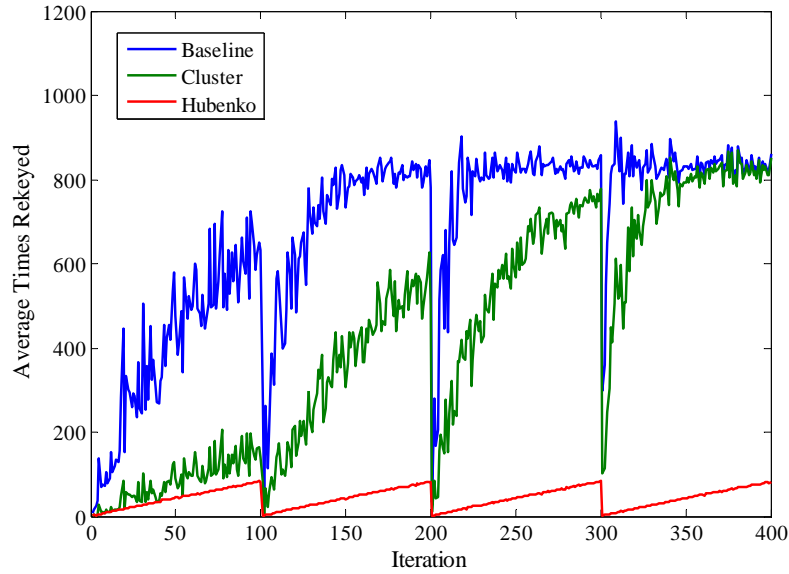
Figure 8. Validation of Architecture Models via Average Rekeys

The results convincingly validate the models of the cluster architecture and the Hubenko architecture used in this study. However, as mentioned in Section 3.10, the baseline architecture evaluated in this study differs from the original baseline architecture because the original study assumed a globally dispersed group of users, whereas this study assumes a local group of UAVs (within the communication range of a Global

Hawk) as the baseline case. Therefore, a direct comparison cannot be made to validate the baseline architecture used in this study.

4.3 Results and Analysis of Performance Metrics

This section interprets and analyzes the relevant data collected from the simulations. The performances of the scenarios tested are analyzed in terms of each metric individually, followed by an overall performance analysis. Results from Scenario 1 are analyzed followed by the results from Scenario 2. Several graphs for each metric are shown in the following sections. The confidence intervals are not shown on the plots, because they are too narrow to be distinguished. However, there are additional plots in Appendices C - I containing the 95% confidence intervals for all data discussed in this chapter.

4.3.1 Analysis of Scenario 1

This section analyzes the results from the Scenario 1 simulations. This scenario represents a UAV swarm where UAVs randomly join the group within the first 15% or 30% of the two hour simulation time. A portion of the swarm (either 25% or 75%) is assigned as highly mobile and a portion of the swarm (either 25% or 75%) departs the group after random intervals. None of the departed UAVs rejoin the group.

4.3.1.1 Analysis of *Total Keys* for Scenario 1

The analysis of *total keys* indicates the scalability of the architecture. The lowest number of keys distributed for a specific combination of factors is important, but the rate at which the keys distributed increases as the swarm size, mobility, joins, and departures increase, is more important in terms of scalability. An examination of the residual plots

of *total keys* reveals that the data does not meet all the criteria to perform an ANOVA. Specifically, the residuals are not normally distributed. However, a logarithmic transformation of the response yields data that does meet the criteria, thus providing a valid ANOVA. Figure 9 shows the log *total keys* versus log swarm size when 25% of the swarm is highly mobile. The four plots in the figure display the data for the four combinations of the join rate and departure rate. For example, the upper left hand plot displays the data when the join rate is set to 15% (all UAVs join the group in the first 15% of the simulation time) and the departure rate is set to 25%. Similarly, Figure 10 displays log *total keys* versus log swarm size when 75% of the swarm is highly mobile. The linear relationship between the log of *total keys* and the log of the swarm size indicates the two are related by a power function.
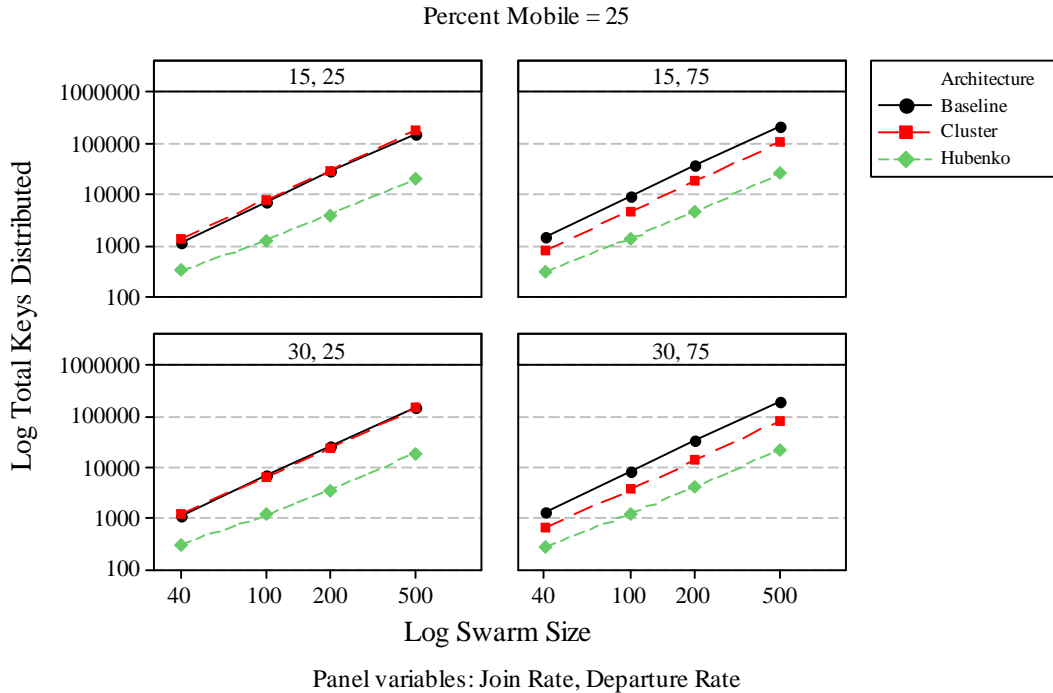

Figure 9. Total Keys versus Swarm Size with 25% Mobility with a Log-Log Scale
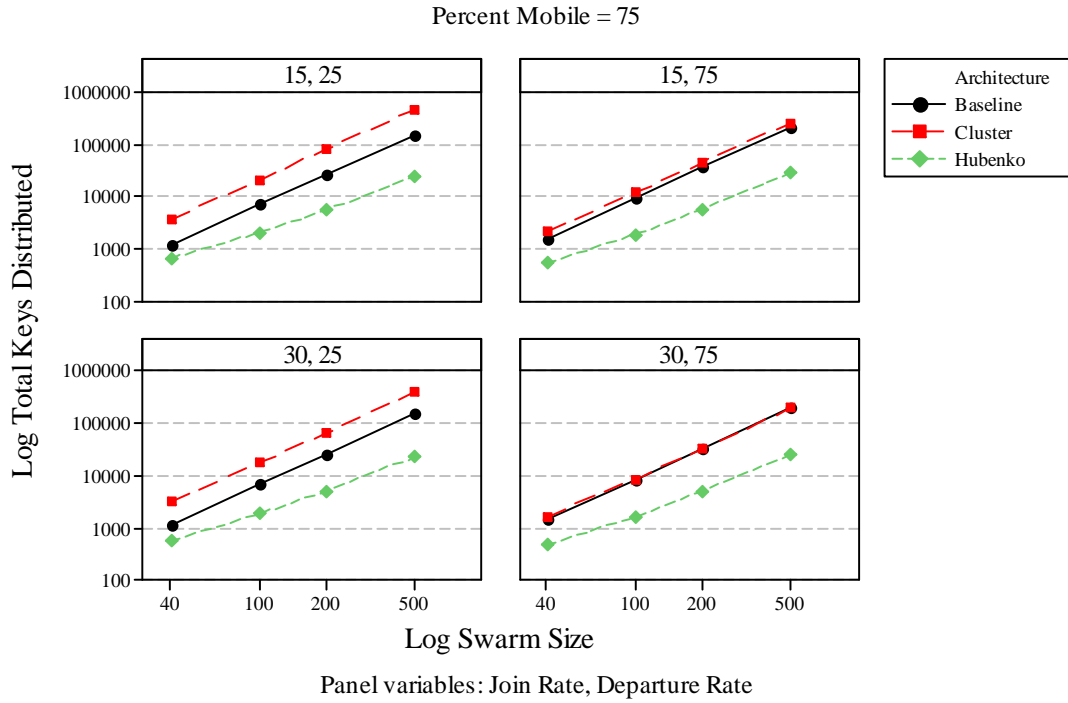
Figure 10. Total Keys versus Swarm Size with 75% Mobility with a Log-Log Scale

As expected, more keys are distributed in the system when the swarm size increases and mobility is high. Also, as predicted, the fewest keys are distributed in the system with the Hubenko architecture. The baseline and cluster architectures performance relative to each other vary depending on the mobility, join rate, and departure rate. By visual inspection it can be seen that the Hubenko architecture has statistically significant differences compared to the baseline and cluster architectures. Also, statistically significant differences can be seen among the various swarm sizes. These differences are verified by the numerical data, which can be found on the detailed plots containing confidence intervals in Appendix D. Though it is difficult to visually see significant statistical differences among the different mobility levels, the differences are evident when examining the confidence intervals. Using the mean response values across

all factors, *total keys* is 86.2% less in the Hubenko architecture compared to baseline and 89.2% less compared to the cluster architecture.

The results of the ANOVA using the log *total keys* are displayed in Table 6. The ANOVA uses the general linear model with the log *total keys* as the response and the swarm size, join rate, departure rate, mobility and architecture as predictors along with their second order interactions. The general linear model assumes the residuals are independent, normally distributed, and have a zero mean. The validity of the ANOVA can be confirmed by visually inspecting the residual plots in Figure 11. The normal probability plot and the histogram show the residuals reasonably fit a normal distribution with a mean of zero. There is some departure from normality in the tails, but the ANOVA is fairly robust with respect to the normality assumption. The versus fits plot shows residuals evenly distributed above and below the center line with no apparent trends indicating the errors are independent.

Table 6. Results of Using an ANOVA on Log Total Keys

| Source of Variation	DF	Adj SS	% Variation	Adj ms	F Ratio	P
SwarmSize	3	414.812	77.587	138.271	161681	0.000
JoinRate	1	0.297	0.056	0.297	347	0.000
DepRate	1	0.421	0.079	0.421	492	0.000
Mobility	1	6.857	1.282	6.857	8017	0.000
Architecture	2	97.979	18.326	48.989	57284	0.000
SwarmSize*JoinRate	3	0.005	0.001	0.002	2	0.097
SwarmSize*DepRate	3	0.046	0.009	0.015	18	0.000
SwarmSize*Mobility	3	0.169	0.032	0.056	66	0.000
SwarmSize*Architecture	6	3.127	0.585	0.521	609	0.000
JoinRate*DepRate	1	0.035	0.007	0.035	41	0.000
JoinRate*Mobility	1	0.007	0.001	0.007	9	0.003
JoinRate*Architecture	2	0.170	0.032	0.085	99	0.000
DepRate*Mobility	1	0.035	0.006	0.035	41	0.000
DepRate*Architecture	2	4.577	0.856	2.289	2676	0.000
Mobility*Architecture	2	5.479	1.024	2.739	3203	0.000
Error	735	0.629	0.117	0.001		
Total	767	534.644	100.000			
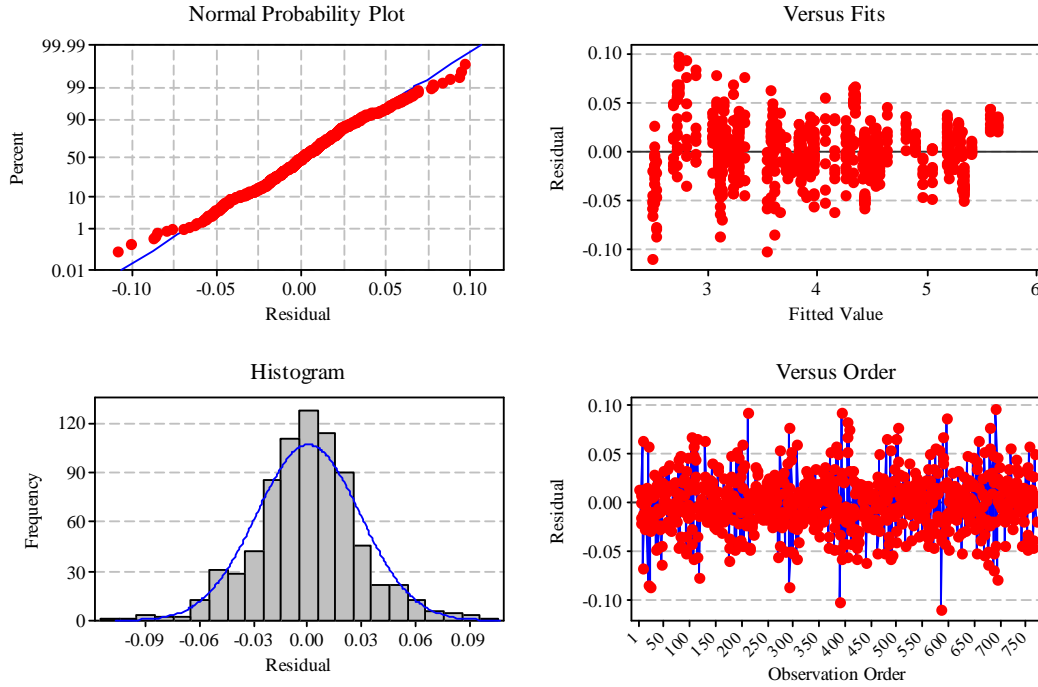
Figure 11. Plots for Verifying the Assumptions of the Log Total Keys ANOVA

As can be verified from Table 6, the model accounts for 99.883% of the variation in the response. All first order terms and all but one of the second order terms have statistical significance at the 0.05 significance level. The swarm size contributes most to variation in the response (77.587%) followed by the architecture (18.326%) and mobility (1.282%). Although, the join rate and departure rate factors are significant, they contribute very little to the overall variation in the response.

The main effects plot for *total keys* is shown in Figure 12. It can be seen that a larger the swarm size and higher mobility increase *total keys*, while the Hubenko Architecture decreases *total keys*. Although the join rate and departure rate have significant effects according to their p-values, they are much smaller in comparison to the other factors, only contributing 0.056% and 0.079% to the variation in the response respectively. Using pair-wise comparisons of the mean responses at the 0.05 level of

significance, each level of the swarm size as well as both levels of mobility have significant statistical differences from all other levels. The Hubenko architecture is statistically different from the cluster and baseline architectures, but the baseline and cluster architectures are not statistically different from each other. The two levels of both the departure rate and the join rate are not statistically different.
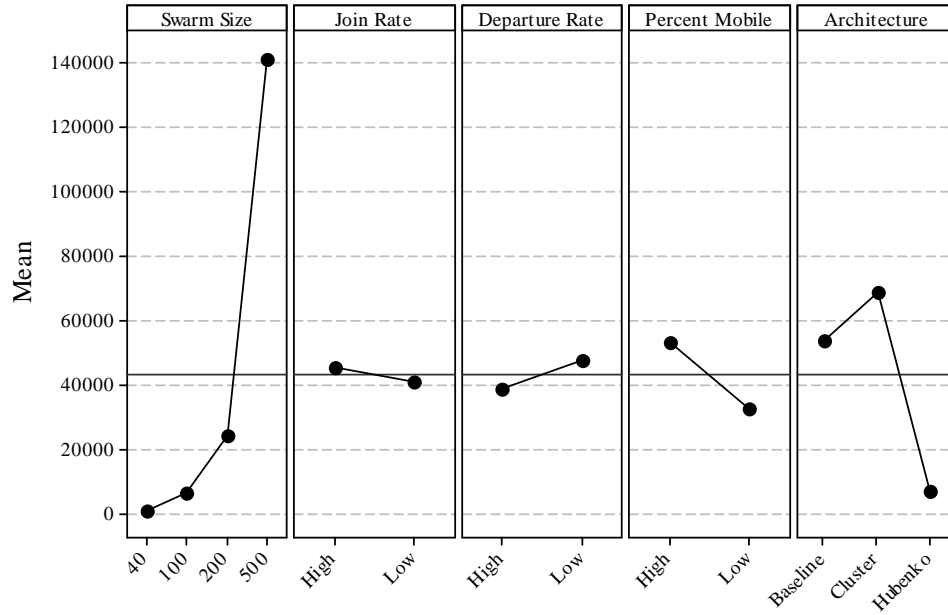

Figure 12. Total Keys Distributed Main Effects Plot

4.3.1.2 Analysis of *Average Rekeys* for Scenario 1

The analysis of *average rekeys* reveals the efficiency of each architecture. When performing an ANOVA on *average rekeys*, the residuals of the response data were found not to fit a normal distribution. Thus, to meet the criteria for a valid ANOVA, a logarithmic transformation of the response data is necessary. Similar to the plots displayed in the previous section, Figure 13 and Figure 14 show the log *average rekeys* versus log swarm size at the 25% and 75% mobility levels respectively.
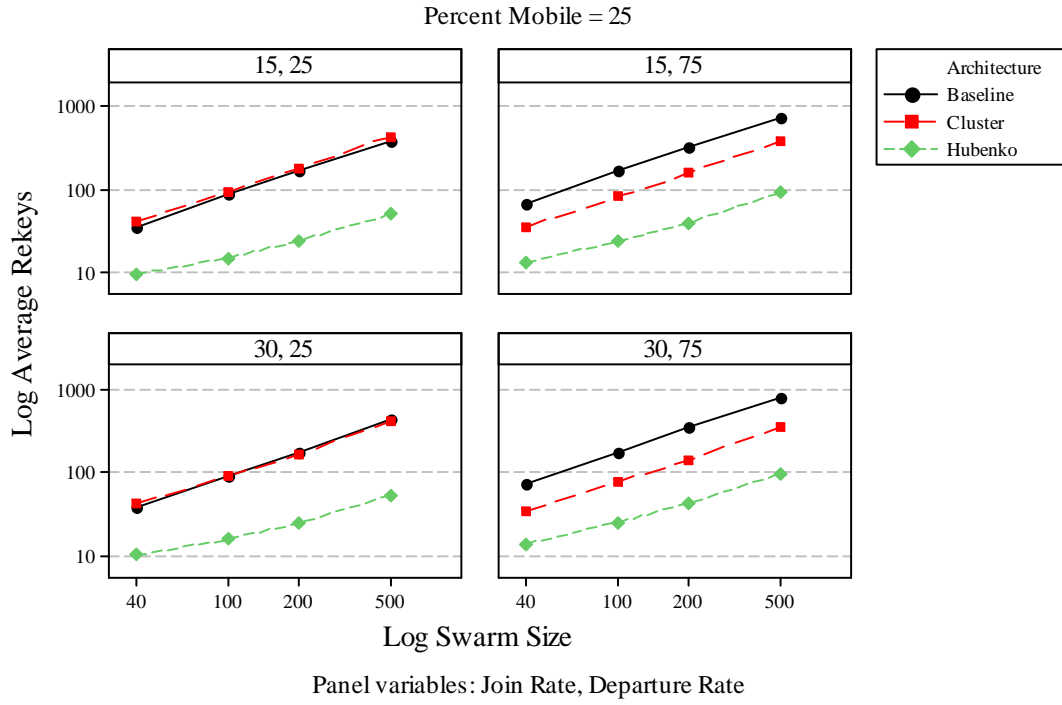
Percent Mobile = 25

Figure 13. Average Rekeys versus Swarm Size with 25% Mobility with a Log-Log Scale
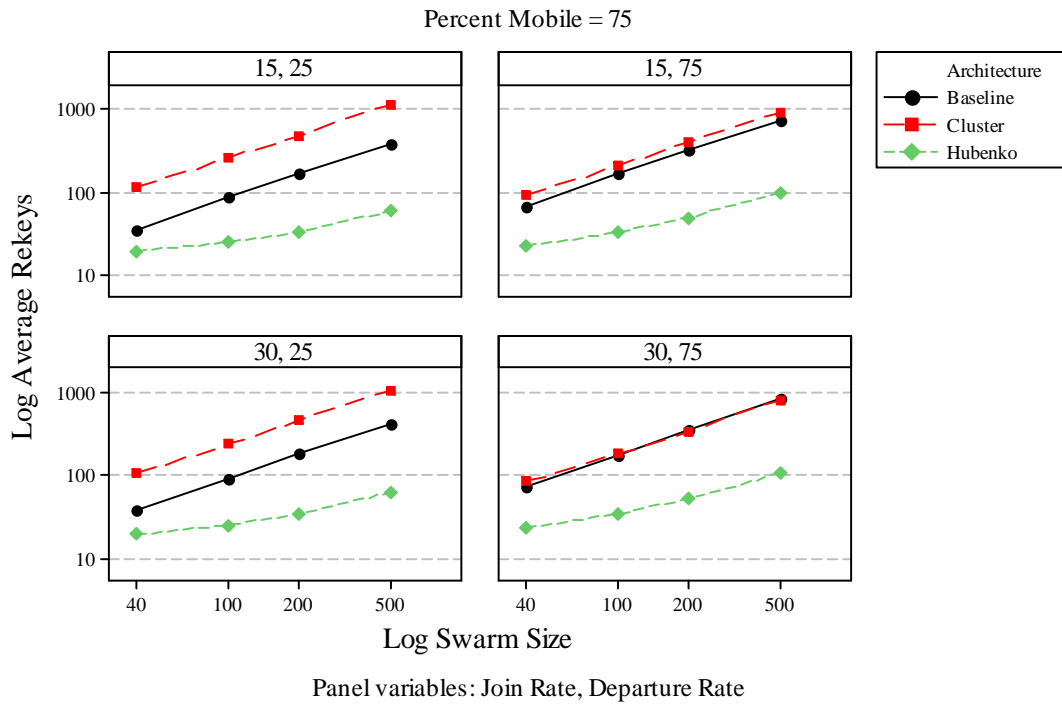
Percent Mobile = 75

Figure 14. Average Rekeys versus Swarm Size with 75% Mobility with a Log-Log Scale

The effects of the factors are similar to those described in the previous section. By visual inspection it can be seen that the Hubenko architecture has significant statistical differences compared to the baseline and cluster architectures and has the lowest *average rekeys* in all factor combinations. Also, significant statistical differences can be seen among the various swarm sizes and the two mobility levels. These differences are verified by the numerical data as well, which can been seen on the detailed plots containing confidence intervals shown in Appendix E. Using the mean response values across all factor levels, *average rekeys* is 84.9% less in the Hubenko architecture compared to the baseline and 87.1% compared to the cluster architecture.

The results of the ANOVA using the log *average rekeys* are displayed in Table 7. The ANOVA uses the general linear model with *average rekeys* as the response and the swarm size, join rate, departure rate, mobility and architecture as predictors along with their second order and some third order interactions.

Table 7. Results of Using an ANOVA on Log Average Rekeys

Source of Variation	DF	Adj SS	% Variation	Adj ms	F Ratio	P
SwarmSize	3	85.686	41.271	28.5622	113181.04	0.000
JoinRate	1	0.049	0.024	0.0490	194.22	0.000
DepRate	1	2.995	1.443	2.9950	11867.90	0.000
Mobility	1	6.742	3.247	6.7423	26717.09	0.000
Architecture	2	97.978	47.191	48.9888	194124.15	0.000
SwarmSize*JoinRate	3	0.002	0.001	0.0007	2.77	0.041
SwarmSize*DepRate	3	0.059	0.029	0.0197	78.23	0.000
SwarmSize*Mobility	3	0.138	0.067	0.0461	182.87	0.000
SwarmSize*Architecture	6	3.127	1.506	0.5211	2065.04	0.000
JoinRate*DepRate	1	0.000	0.000	0.0004	1.57	0.211
JoinRate*Mobility	1	0.004	0.002	0.0042	16.62	0.000
JoinRate*Architecture	2	0.170	0.082	0.0851	337.22	0.000
DepRate*Mobility	1	0.046	0.022	0.0463	183.55	0.000
DepRate*Architecture	2	4.577	2.205	2.2887	9069.44	0.000
Mobility*Architecture	2	5.485	2.642	2.7423	10866.80	0.000
SwarmSize*JoinRate*Mobility	3	0.002	0.001	0.0008	2.97	0.031
SwarmSize*JoinRate*Architecture	6	0.006	0.003	0.0010	3.97	0.001
SwarmSize*DepRate*Architecture	6	0.106	0.051	0.0176	69.79	0.000
SwarmSize*Mobility*Architecture	6	0.231	0.111	0.0385	152.52	0.000
JoinRate*DepRate*Architecture	2	0.006	0.003	0.0032	12.72	0.000
JoinRate*Mobility*Architecture	2	0.004	0.002	0.0018	7.06	0.001
DepRate*Mobility*Architecture	2	0.027	0.013	0.0134	53.12	0.000
Error	708	0.179	0.086	0.0003		
Total	767	207.620				

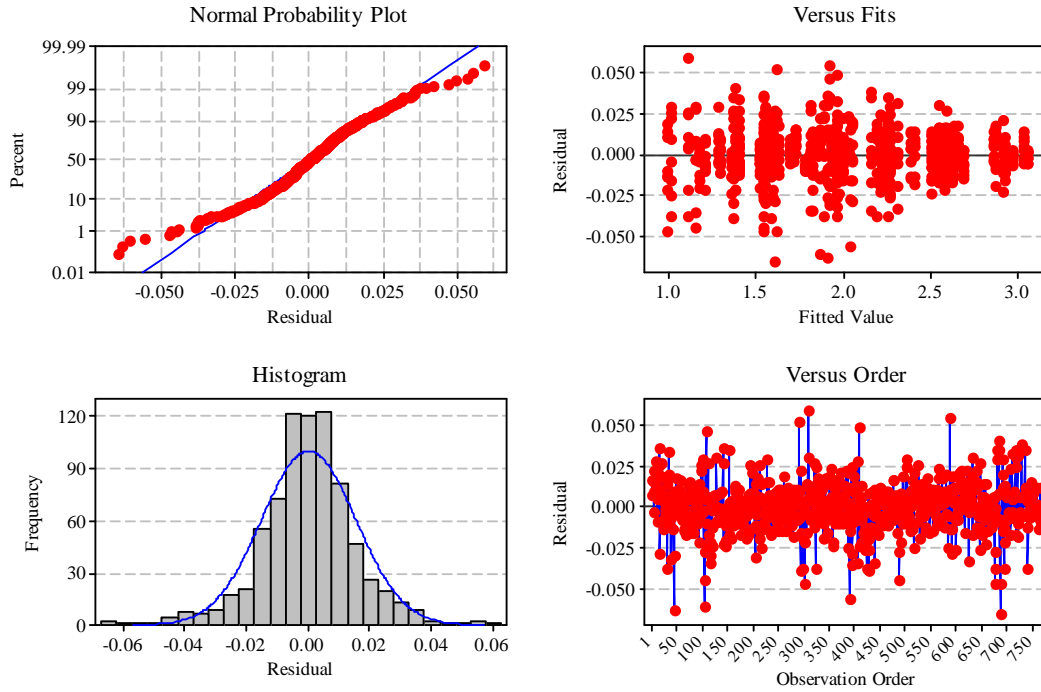The validity of the ANOVA can be confirmed by visually inspecting the residual plots in Figure 15.

Figure 15. Plots for Verifying the Assumptions of the Log Average Rekeys ANOVA

Table 7 indicates that the computed model accounts for 99.914% of the variation in *average rekeys* with all first order, and many second and third order terms having statistical significance at the 0.05 significance level. For this response the architecture contributes most to the variation (47.191%), followed by the swarm size (41.271%). The mobility (3.247%) contributes the third most to the variation in the response.

The main effects plot for the average rekeys per UAV is shown in Figure 16. A larger swarm size and higher mobility increases *average rekeys*, while the Hubenko Architecture decreases *average rekeys*, which confirms the findings from the plots above. The join rate has little effect, while a higher departure rate suggests higher *average rekeys*. Using pair-wise comparisons of the mean responses at the 0.05 level of

significance, each level of the swarm size as well as both levels of mobility have significant statistical differences from all other levels. The Hubenko architecture is statistically different from the cluster and baseline architectures, but the baseline and cluster architectures are not statistically different from each other. The two levels of the join rate are not statistically different. Although it appears departure rate levels are statistical different the pair-wise comparison does not confirm this.
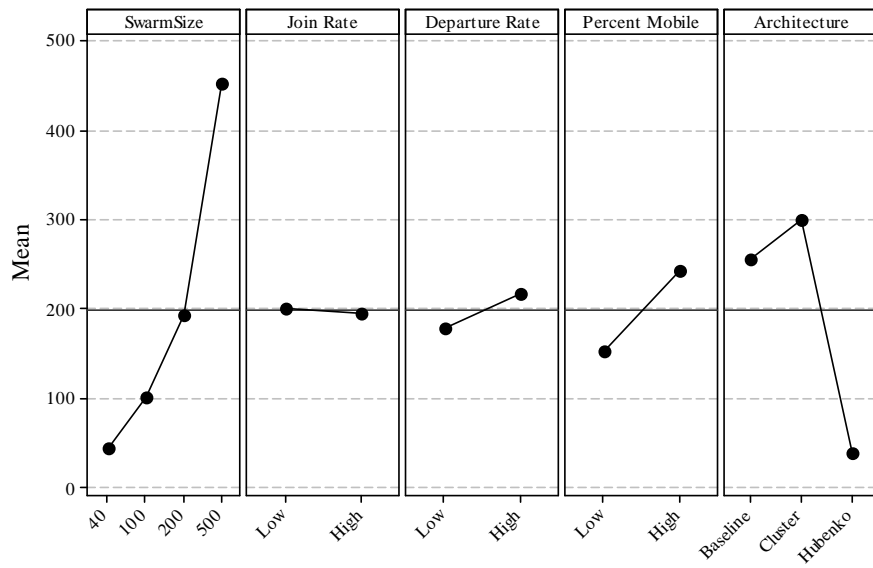

Figure 16. Average Rekeys per UAV Main Effects Plot

4.3.2 Analysis of Scenario 2

This section analyzes the results from the Scenario 2 simulations. Since the join rate has little effect on the responses compared to the other factors and the departure rate is not applicable because there are continuous departures, only three factors (swarm size, mobility, and architecture) are varied. However, more levels are added to the swarm size and swarm mobility factors, and two additional metrics are measured (*average bandwidth*

and *battery consumed*). The calculations used to measure *average bandwidth* and *battery consumed* can be found in Appendices A and B respectively.

Recall that, Scenario 2 represents a swarm of UAVs that needs to be sustained for a prolonged period of time. Thus, this scenario simulates UAVs joining, departing, and rejoining the swarm over a 12 hour period. Each individual UAV's join, departure, and rejoin times are based on random variables and a randomly assigned battery life ranging from 30 minutes to 3 hours. The percentage of UAVs assigned as highly mobile is the same as Scenario 1, but 50% and 90% mobility levels are tested as well.

4.3.2.1 Analysis of *Total Keys* for Scenario 2

As in Scenario 1, an examination of the residual plots of *total keys* reveals that the data does not meet all the criteria to perform an ANOVA. Specifically, the residuals do not fit a normal distribution. However, a logarithmic transformation of the response yields data that does meet the criteria, thus providing a valid ANOVA. Figure 17 contains four plots displaying the log *total keys* versus log swarm size. Each plot represents a different mobility level. For example, the plot in the upper left hand corner contains data from simulations run with the 25% of the swarm being highly mobile. As with the rest of the Scenario 2 plots, the confidence intervals are not shown because they are too narrow to be distinguishable. Additional plots containing 95% confidence intervals can be found in Appendix F.

As expected, *total keys* increases as the swarm size increases and the fewest keys are distributed in the Hubenko architecture. It is evident that there are statistically significant differences among the three architectures and the various swarm sizes. These

differences are also verified by the numerical data displayed on the graphs in Appendix F.
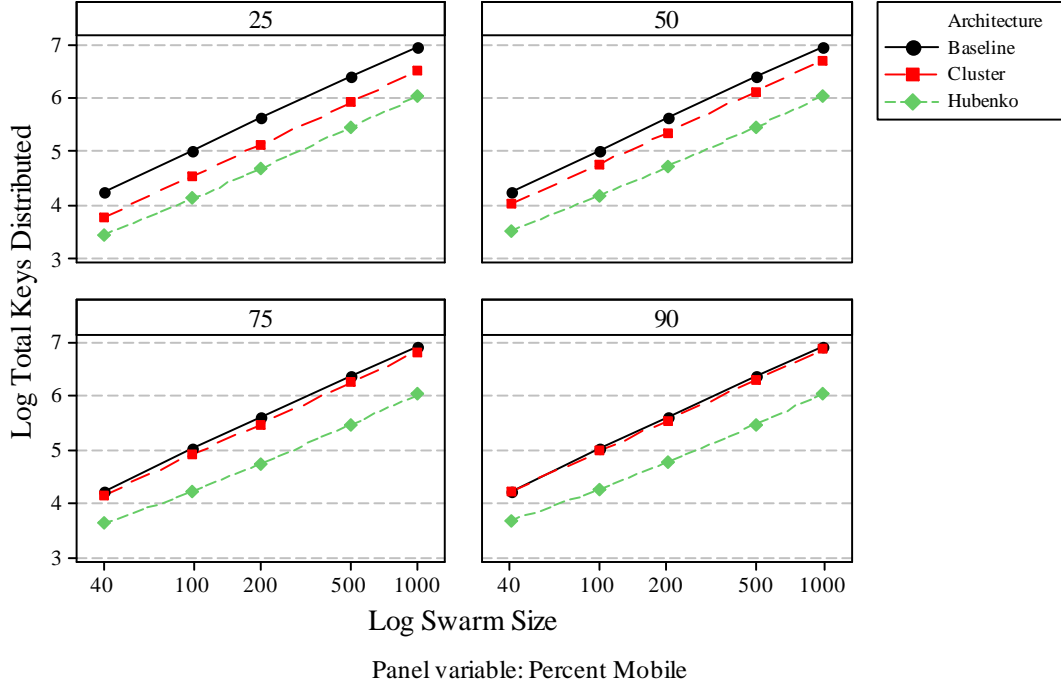

Figure 17. Total Keys versus Swarm Size with a Log-Log Scale

Unlike Scenario 1, the cluster architecture outperforms the baseline architecture in every situation which reveals the negative impact departures and rejoins have on the baseline architecture. The cluster architecture and Hubenko architecture are better designed to handle this. The mobility rate does appear to affect the cluster architecture, however only the 25% and 90% levels are significantly different when the data is left ungrouped. The mobility levels are significantly different from all other mobility levels when the data is grouped by architecture and swarm size for both the cluster and the Hubenko architecture. Using the mean response values across all factors, 87.3% less keys are distributed in the Hubenko architecture compared to the baseline, and 80.5% less keys are distributed compared to the cluster architecture.

The results of the ANOVA using the log *total keys* are displayed in Table 8. The validity of the ANOVA can be confirmed by visually inspecting the residual plots in Figure 18.

Table 8. Results of Using an ANOVA on Log Total Keys

| Source of Variation	DF	Adj SS	% Variation	Adj MS	F Ratio	P
Swarm Size	4	1041.807	86.21	260.452	1850673.19	0.000
Mobility	3	5.252	0.43	1.751	12438.89	0.000
Architecture	2	153.14	12.67	76.570	544079.53	0.000
Swarm Size*Mobility	12	0.147	0.01	0.012	87.05	0.000
Swarm Size*Architecture	8	2.035	0.17	0.254	1807.63	0.000
Mobility*Architecture	6	5.725	0.48	0.954	6780.28	0.000
Swarm Size*Mobility*Architecture	24	0.197	0.02	0.008	58.25	0.000
Error	1140	0.16	0.01	0.000		
Total	1199	1208.463				
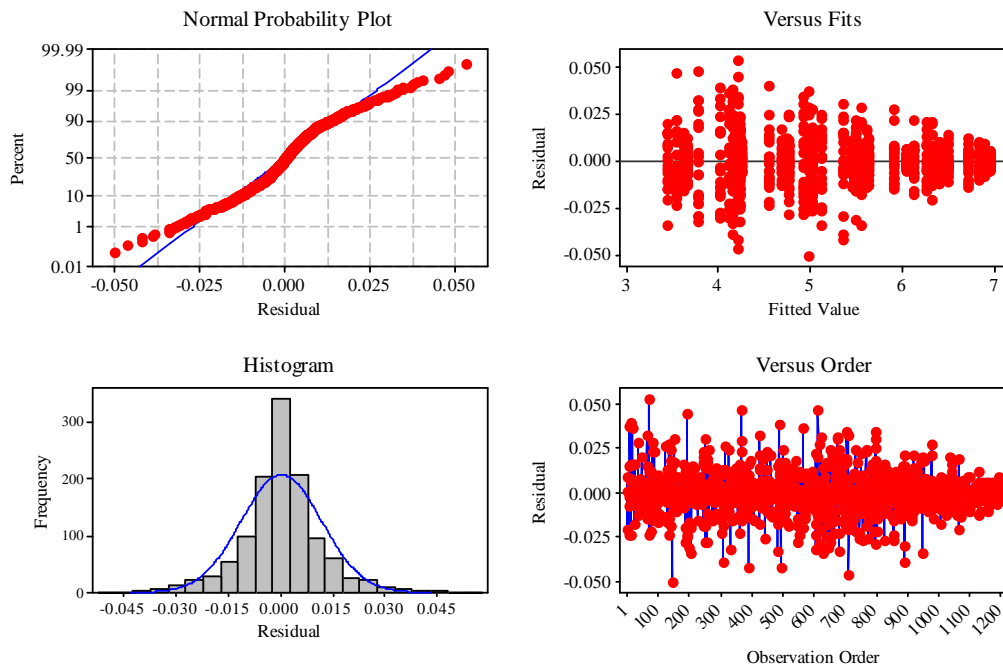
Figure 18. Plots for Verifying the Assumptions of the Log Total Keys ANOVA

As can be verified from Table 8, the model accounts for 99.99% of the variation in the log *total keys* with all first, second, and third order terms having statistical significance at the 0.05 significance level. The swarm size contributes most to variation in the response (86.21%) followed by the architecture (12.67%). Although all other

terms are significant according to the model, their percent variation in the response is less than 1% each.

The main effects plot for *total keys* is shown in Figure 19. A larger swarm size and higher mobility increases *total keys*, while the Hubenko Architecture decreases *total keys*. A pair-wise comparison of the mean responses at the 0.05 level of significance shows each level of the swarm size has significant statistical differences from all other levels. Each architecture is also significantly different from all other architectures. Although the mean increases as the percentage of highly mobile users increases, none of the mobility rate levels are statistically different from any other levels when the data is left ungrouped.
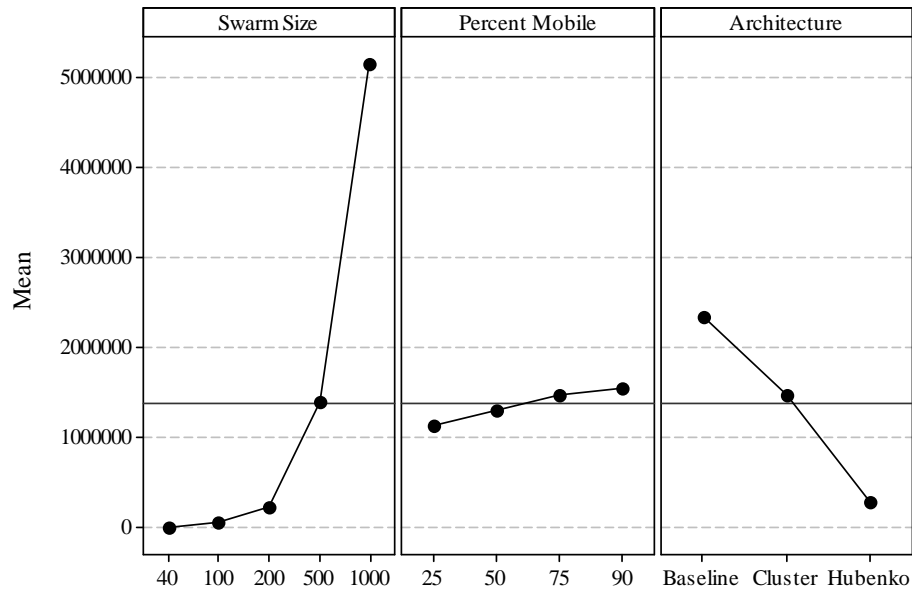

Figure 19. Main Effects Plot for Total Keys

4.3.2.2 Analysis of *Average Rekeys* for Scenario 2

An examination of the residual plots of *average rekeys* reveals that the data does not meet all the criteria to perform an ANOVA. Specifically, the residuals of the response data do not fit a normal distribution. Thus, to meet the criteria to perform a

valid ANOVA a logarithmic transformation of the response data is necessary. **Error! Reference source not found.** contains four plots displaying the log *average rekeys* versus log swarm size, with each plot representing a different mobility level. Additional plots containing 95% confidence intervals can be found in Appendix G.
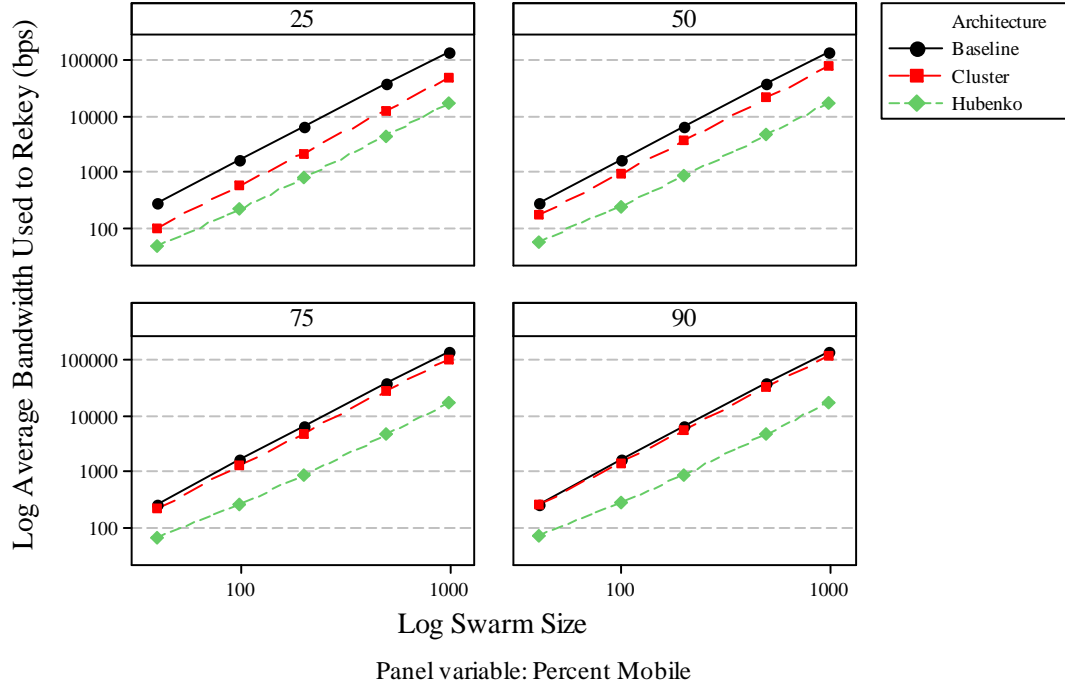

Figure 20. Average Rekeys versus Swarm Size with Log-Log Scale

Statistically significant differences can be seen among the three architectures and the various swarm sizes. These differences are verified by the numerical data from Appendix G. According to the numerical data, a UAV in the Hubenko architecture rekeys an average of 87.3% less than a UAV in the baseline architecture. Similarly, a UAV in the Hubenko architecture rekeys an average of 79.9% less than a UAV in the cluster architecture.

The results of the ANOVA using the log *average rekeys* are displayed in Table

Table 9. The assumptions of the ANOVA can be confirmed by visually inspecting the

residual plots in Figure 21.

Table 9. Results of Using an ANOVA on Log Average Rekeys

| Source of Variation	DF	Adj SS	% Variation	Adj MS	F Ratio	P
Swarm Size	4	228.2673	57.79	57.0668	229498.16	0.000
Mobility	3	5.2682	1.33	1.7561	7062.09	0.000
Architecture	2	153.0614	38.75	76.5307	307773.42	0.000
Swarm Size*Mobility	12	0.1549	0.04	0.0129	51.90	0.000
Swarm Size*Architecture	8	2.0506	0.52	0.2563	1030.83	0.000
Mobility*Architecture	6	5.7220	1.45	0.9537	3835.25	0.000
Swarm Size*Mobility*Architecture	24	0.1955	0.05	0.0081	32.75	0.000
Error	1140	0.2835	0.07	0.000		
Total	1199	395.0033				
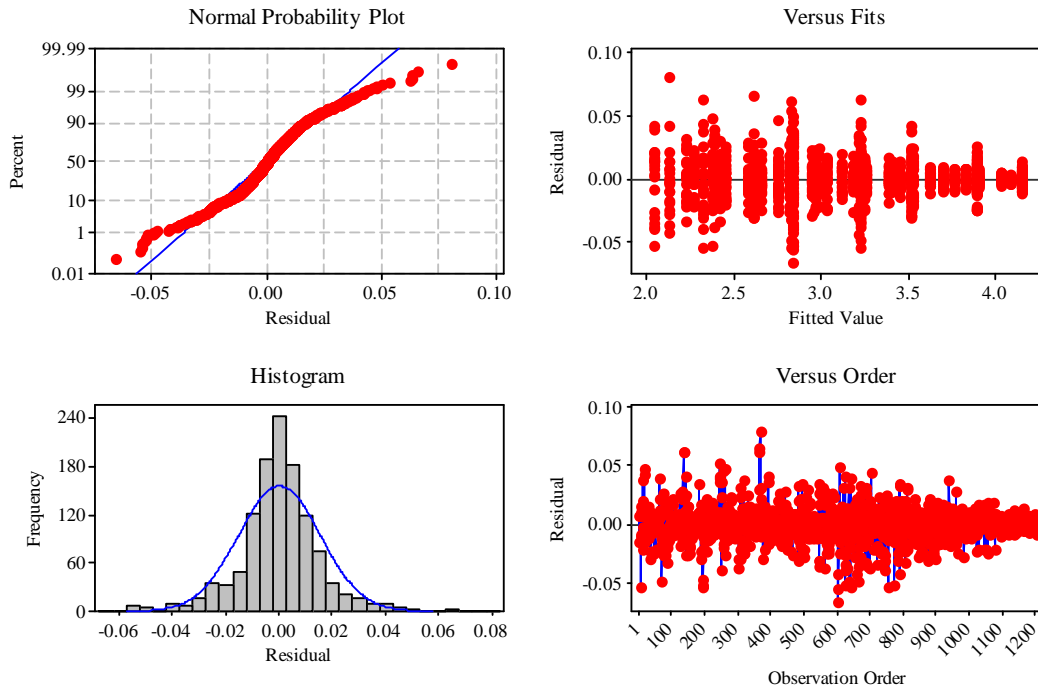
Figure 21. Plots for Verifying the Assumptions of the Log Average Rekeys ANOVA

As can be verified from Table 9, the model accounts for 99.93% of the variation

in the log *average rekeys* with all first, second, and third order terms having statistical

significance at the 0.05 significance level. The swarm size contributes most to variation

in the response (57.79%) followed by the architecture (38.75%), the second order

interaction between mobility and architecture (1.45%), and mobility (1.33%). Although significant, all other terms contribute less than 1% each to the variation in the response.

The main effects plot for *average rekeys* is shown in Figure 22. A larger swarm size and higher mobility increases *average rekeys*, while the Hubenko architecture reduces *average rekeys*. Using pair-wise comparisons of the mean responses at the 0.05 level of significance, each level of the swarm size has significant statistical differences from all other levels. Each architecture also has significant statistical differences from all other architectures. The 25% mobility level and 90% mobility level are the only mobility levels with significant statistical differences.
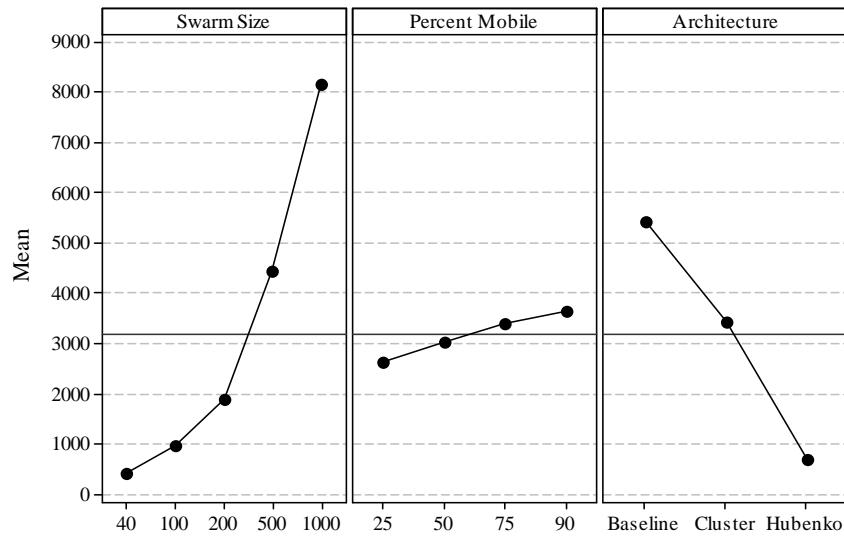

Figure 22. Main Effects Plot for Average Rekeys Per UAV

4.3.2.3 Analysis of *Average Bandwidth* for Scenario 2

To meet the criteria to perform a valid ANOVA a logarithmic transformation of the response data is necessary. Figure 23 contains four plots displaying the log *average bandwidth* versus log swarm size, with each plot representing a different mobility level.

71

This displays the linear relationship between log *average bandwidth* and log swarm size. Additional plots containing 95% confidence intervals are found in Appendix H. The formulas and assumptions used to calculate *average bandwidth* are presented in Appendix A.
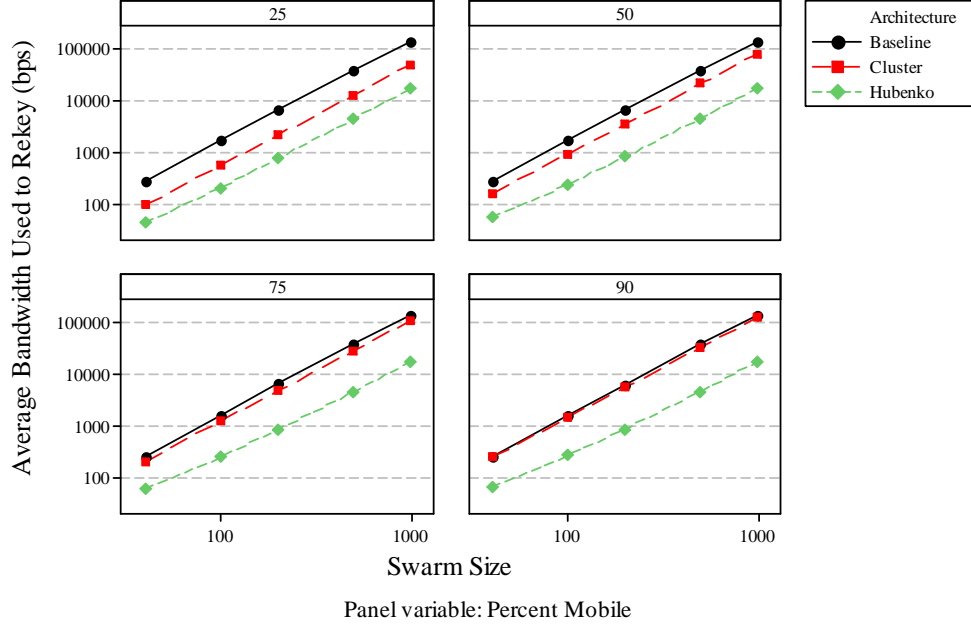

Figure 23. Average Bandwidth versus Swarm Size with Log-Log Scale

The plots indicate a significant statistical difference among the three architectures and the various swarm sizes. These differences are verified by the numerical data, which can be found on the plots in Appendix H. Without grouping by the other factors, none of the mobility levels are statistically different.

In terms of reducing the use of limited resources, such as bandwidth, the power of the Hubenko architecture is evident. At the 25% mobility level both the cluster and Hubenko architecture scale well, relative to the baseline, as the swarm size increases. However, once mobility increases, the bandwidth used by the cluster architecture nears that of the baseline, while the Hubenko architecture is minimally affected. At the 90%

mobility level the Hubenko architecture uses an average of 85.3% less *average*

bandwidth than the cluster architecture and 87.3% less than the baseline architecture.

The results of the ANOVA using the log *average bandwidth* are displayed in

Table 10. The validity of the ANOVA is confirmed by visually inspecting the residual

plots in Figure 24.

Table 10. Results of Using an ANOVA on Log Average Bandwidth

| Source of Variation	DF	Adj SS	% Variation	Adj MS	F Ratio	P
SwarmSize	4	1041.807	86.21	260.455	1850673.19	0.000
Mobility	3	5.252	0.43	1.751	12438.89	0.000
Architecture	2	153.137	12.67	76.568	544079.53	0.000
SwarmSize*Mobility	12	0.147	0.01	0.012	87.05	0.000
SwarmSize*Architecture	8	2.036	0.17	0.254	1807.63	0.000
Mobility*Architecture	6	5.725	0.48	0.954	6780.28	0.000
SwarmSize*Mobility*Architecture	24	0.197	0.02	0.008	58.25	0.000
Error	1140	0.161	0.01	0.000		
Total	1199	1208.463				
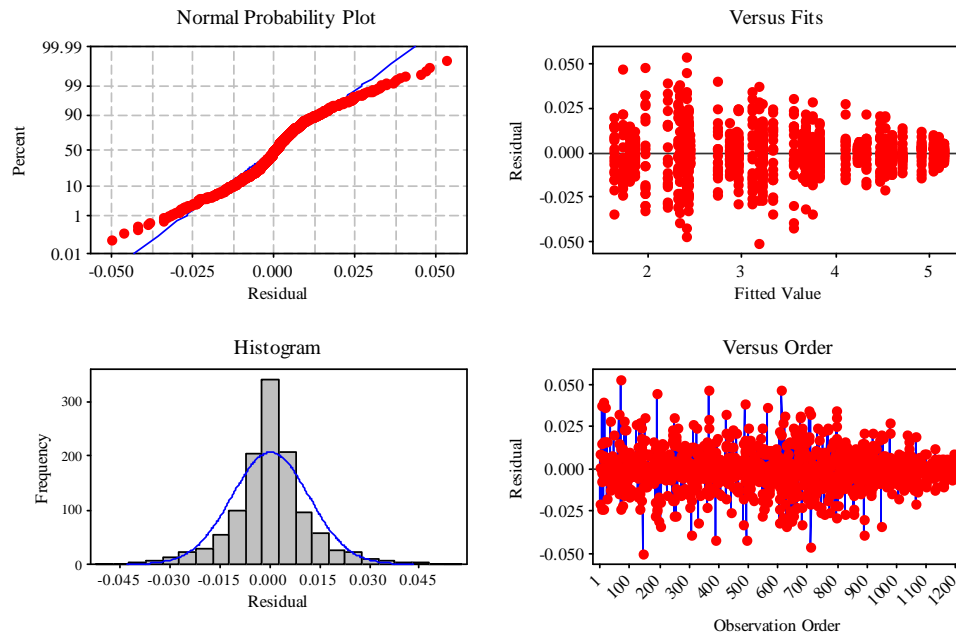
Figure 24. Plots for Verifying the Assumptions of the Log Average Bandwidth ANOVA

The model accounts for 99.99% of the variation in the log *average bandwidth*

with all first, second, and third order terms having statistical significance at the 0.05

significance level. The swarm size contributes most to variation in the response

(86.21%) followed by the architecture (12.67%). Although all other terms are significant according to the model, their percent variation in the response is less than 1% each.

The main effects plot for *average bandwidth* is shown in Figure 25. Here again, a larger swarm size and higher mobility increases *average bandwidth*, while the Hubenko architecture reduces the amount of bandwidth used. Using pair-wise comparisons of the mean responses at the 0.05 level of significance, each level of the swarm size has significant statistical differences from all other levels. Each architecture is significantly different from all other architectures. None of the mobility rate levels are statistically different from any of the other levels when the data is left ungrouped. However, when grouped by architecture and swarm size, each mobility level for the Hubenko and cluster architecture has significant statistical differences from all other levels as verified in each of the plots in Appendix H.
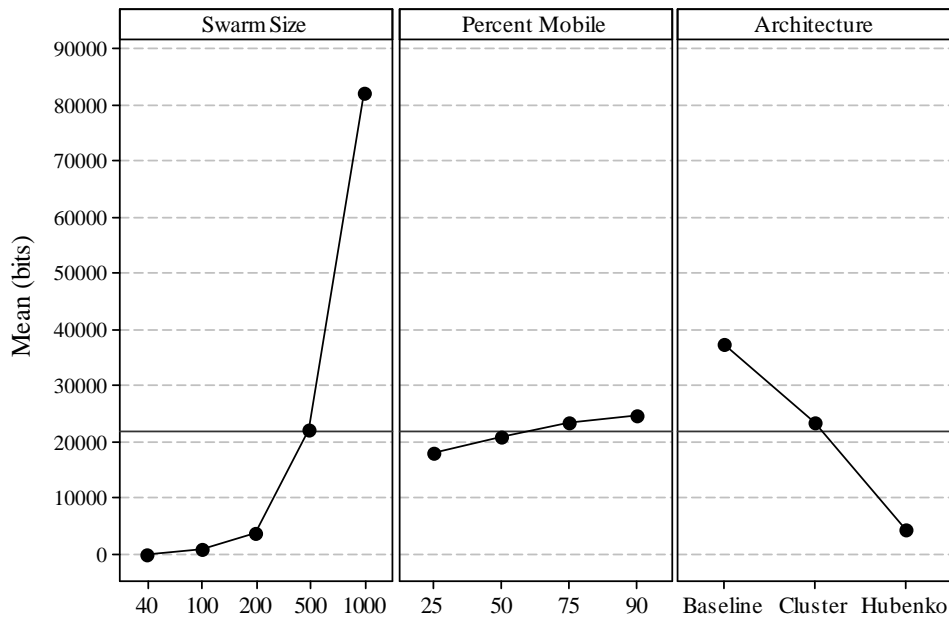

Figure 25. Main Effects Plot for Average Bandwidth

74

4.3.2.4 Analysis of Average Percentage of Battery Consumed to Rekey

Figure 26 contains four plots displaying *battery consumed* versus the swarm size, with each plot representing a different mobility level. Additional plots containing 95% confidence intervals can be found in Appendix I. The formulas and assumptions used to calculate *battery consumed* are presented in Appendix B.
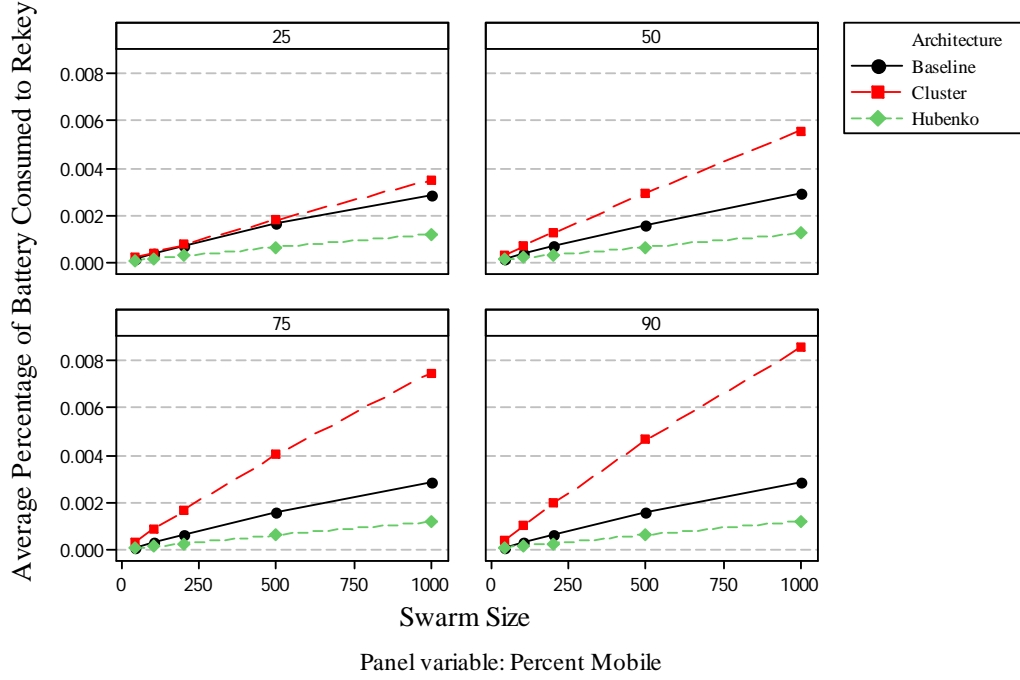

Figure 26. Battery Consumed versus Swarm Size

The plots indicate a statistically significant difference among the three architectures and the various swarm sizes. These differences are verified by the numerical data displayed in the graphs in Appendix I. Aside from differences in the cluster architecture, it is difficult to visually determine differences in the mobility levels. Interestingly, the baseline architecture outperforms the cluster architecture in terms of the response. In the baseline architecture the Global Hawk uses fuel, not batteries and rekeys all of the swarm members. Thus, battery is only consumed when a swarm member

receives a new key. However, in the cluster and Hubenko architectures, the keys are distributed by cluster leaders, which are swarm members themselves, and thus battery is consumed to both transmit and receive a key.

Although the results appear insignificant because the percentage of battery consumed is so small, the relative performance differences among the architectures are very significant. Not included in the simulation are routing, lost packets, and the higher level protocols that provide reliability. Thus, the simplest case is assumed to rekey the swarm: one packet transmitted to, and received by each swarm member containing the key. When routing and reliable protocols are factored into the experiments *battery consumed* will undoubtedly increase. Thus, the rate at which the percentage of battery consumed increases as the swarm size and mobility increase provides more useful information. Figure 26 shows the growth rate of the response versus swarm size and mobility is the lowest in the Hubenko architecture.

The results of the ANOVA using *battery consumed* are displayed in Table 11. The validity of the ANOVA can be confirmed by visually inspecting the residual plots in Figure 27.

Table 11. Results of Using an ANOVA on Battery Consumed

Source of Variation	DF	Adj SS	% Variation	Adj MS	F Ratio	P
Swarm Size	4	0.0017230	48.33	0.0004308	615782.46	0.000
Mobility	3	0.0000844	2.37	0.0000281	40216.68	0.000
Architecture	2	0.0007733	21.69	0.0003866	552714.19	0.000
Swarm Size*Mobility	12	0.0000611	1.71	0.0000051	7278.48	0.000
Swarm Size*Architecture	8	0.0006470	18.15	0.0000809	115607.43	0.000
Mobility*Architecture	6	0.0001555	4.36	0.0000259	37039.88	0.000
Swarm Size*Mobility*Architecture	24	0.0001202	3.37	0.0000050	7158.87	0.000
Error	1140	0.0000008	0.02	0.0000000		
Total	1199	0.0035652				

The model in Table 11 accounts for 99.98% of the variation in *battery consumed* with all first, second, and third order terms having statistical significance at the 0.05 significance level. All factors and their interactions, aside from error, account for at least

one percent of the variation in the response. The swarm size contributes most to variation in the response (48.33%) followed by the architecture (21.69%) and the second order interaction between swarm size and architecture (18.15%).
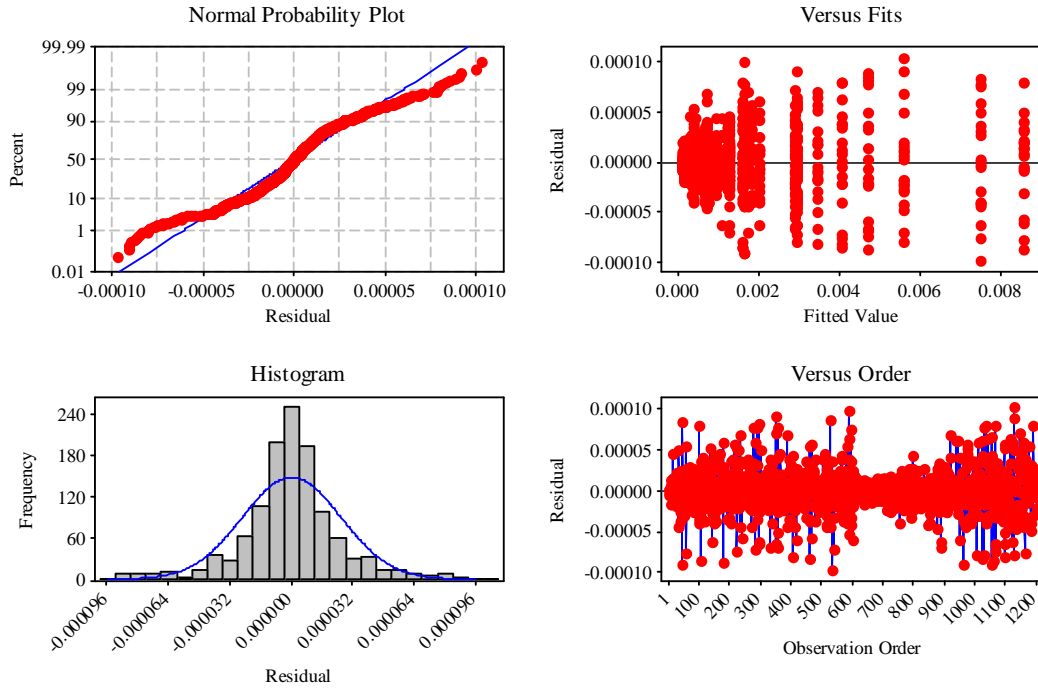

Figure 27. Plots for Verifying the Assumptions of the Battery Consumed ANOVA

The main effects plot for *battery consumed* is shown in Figure 28. It can be seen a larger swarm size and higher mobility increases *battery consumed*, while the Hubenko Architecture decreases *battery consumed*. Using pair-wise comparisons of the mean responses at the 0.05 level of significance, each level of the swarm size has significant statistical differences from all other levels. Also, each architecture has significant statistical differences from all other architectures. The 25% mobility level is the only mobility level with significant statistical differences from all other levels.
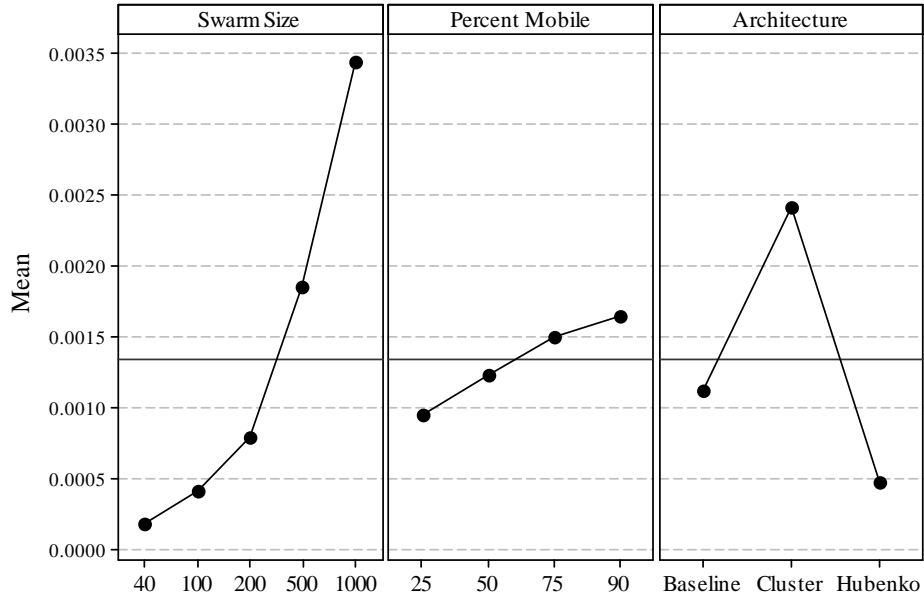
Figure 28. Main Effects Plot for Battery Consumed

4.3 Overall Analysis

Several conclusions can be drawn from the simulations conducted. Most importantly, the statistical analysis of the data confirms the hypothesis. The Hubenko architecture provides statistically significant performance gains over the commonly used baseline and cluster multicast security architectures. By taking advantage of spatial clustering to decrease the negative performance impact of joins and departures, and integrating GACA-GKM to decrease the negative performance impact of highly mobile UAVs, the Hubenko architecture outperforms the baseline and cluster architectures in all of the conducted experiments. Using the data from both scenarios, the following summarizes the performance gains achieved by the Hubenko architecture compared to the baseline architecture (includes the smallest and highest gains across all configurations):

- 57.8 - 87.6% less total keys distributed

- 59.6 - 87.9% less rekeys per UAV

- 73.0 - 87.9% less bandwidth used to rekey

- 16.9 - 58.8% less battery consumed to rekey

Similarly, the following summarizes the performance gains achieved by the Hubenko architecture compared to the cluster architecture (ranging from the smallest to the highest gains across all configurations):

- 55.2 - 94.9% less total keys distributed

- 59.0 – 94.8% less rekeys per UAV

- 55.2 – 85.4% less bandwidth used to rekey

- 54.3 – 85.4% less battery consumed to rekey

It is also important to realize these performance gains coincide with an overall improvement in the security of the system via GACs and independent SEKs for each cluster.

Other conclusions that can be drawn from the overall analysis of the simulations are the significance and effects of the factors. First, comparing the data from the two scenarios, it can be seen that the longer simulation time, and the ability of UAVs to continuously depart and rejoin the swarm significantly increases *total keys* and *average rekeys*. For example, if the configurations from Scenario 1 and Scenario 2 are compared when the swarm size is 500 and mobility is at 75%, *total keys* in the baseline case increases over 14 times from Scenario 1 to Scenario 2. After accounting for the difference in simulation length by dividing *total keys* by the number of hours, *total keys* in the baseline increase 2.02 times per hour. The Hubenko and cluster architectures see

similar, but smaller, increases as well (about a 1.86 and 1.35 times per hour increase respectively). Similar trends can also be seen for *average rekeys*.

As expected, the swarm size significantly contributes to the variation in all of the responses, causing the most variation in all but one of the measured responses. The architecture is the second largest contributing factor in all of the responses, except for one, where it is the largest. As discussed previously, the join rate is significant according to the p-value from the general linear model, but it contributes very little to the variation in the measured responses. The mobility of the swarm has no effect on the baseline architecture, but has significant effects in both the Hubenko and cluster architectures.

4.4 Summary

This chapter presents and analyzes the data collected from the simulations of three different security architectures applied a swarm of autonomous UAVs. The validation of the simulated architecture models is presented followed by a statistical analysis of both scenarios in terms of each performance metric. Finally, an overall analysis and discussion of the results is provided.

V. Conclusions and Recommendations

5.1 Introduction

This chapter summarizes the overall conclusions of the research. Section 5.2 presents the conclusions from the experimental results. The significance of this research is discussed in Section 5.3. Finally, Section 5.4 describes recommendations for areas of future research.

5.2 Conclusions of Research

The Hubenko architecture can successfully be applied to a swarm of autonomous UAVs. Furthermore, the Hubenko architecture significantly outperforms the two other security architectures studied in terms of reducing *total keys*, *average rekeys*, *average bandwidth*, and *battery consumed*. By taking advantage of spatial clustering to decrease the negative performance impact of joins and departures, and integrating GACA-GKM to decrease the negative performance impact of highly mobile UAVs, the Hubenko architecture is a very efficient and scalable architecture.

The largest performance gains are seen in large, highly mobile swarms, in which UAVs continuously join and depart the group. In this type of environment the Hubenko architecture reduces the total keys distributed, average rekeys per UAV, average bandwidth used to rekey up to 88% compared to the baseline architecture. The average percentage of battery consumed is reduced up to 59% compared to the baseline.

In most cases, statistical analysis of the metrics found swarm size to be the largest factor contributing to the variation in the responses, followed by the architecture, and mobility. The join rate and departure rate significantly affect the response but contribute little to the variation in the response relative to the other factors.

5.3 Significance of Research

This research is the first to provide a practical and direct application of the Hubenko architecture, which was broadly designed for securing group communication in the GIG. It is also among the first to address secure group communication in an autonomous UAV swarm. Through careful research and simulation, this study shows the Hubenko architecture is not only a viable solution to securing group communication in an autonomous UAV swarm, but a very efficient and scalable solution as well. With this added security component, a swarm of autonomous UAVs can provide a unique and powerful net-centric asset to support the warfighter. This research also tests the Hubenko architecture under more realistic scenarios, further validating its potential as an efficient and scalable group communication security architecture.

5.4 Recommendations for Future Research

The Hubenko architecture should be tested in environments such as OPNET where network constraints such as delay, packet loss, and retransmissions can be accurately modeled. In addition multicast ad hoc routing protocols should be investigated to find the best performance for specific scenarios.

Recent research efforts have developed mobility models such as the random waypoint model to simulate UAV swarms performing a search [CaB02]. Applying the Hubenko architecture to those studies would be useful. In addition to the more realistic model of swarm's mobility, the performance impact of the Hubenko architecture on the search could be evaluated.

This study assumed a simple pair-wise rekeying distribution protocol. However, any rekeying protocol could be used with the Hubenko architecture. Evaluating the best

rekeying distribution protocol, such as pair-wise rekeying, distributed, secure lock, and hierarchical trees in an autonomous UAV swarm performing a search would be very beneficial.

Finally, research could further expand the scenarios herein to include other assets in the GIG, which communicate with a UAV swarm. These include ground troops, wireless sensor networks, satellites, or other air assets. This would call for more levels in the hierarchy of the Hubenko architecture and would provide an interesting test of the robustness of the architecture.

5.5 Summary

This chapter presented the conclusions of this research. The significance of the research was discussed as well as several recommendations for future research.

Appendix A. Bandwidth Used to Rekey Calculations

Bandwidth is a limited resource in MANETs such as a swarm of UAVs. Although maintaining security of the swarm is an important necessity, it is not usually the primary function of the swarm's communication and should therefore not dominate the bandwidth. Thus, bandwidth used to rekey is an important metric when considering which security architecture to implement.

A specific encryption scheme or algorithm has not been selected, as it is beyond the scope of this work. The best encryption scheme depends on the security needs of the application and the capabilities of the UAVs. However, for the purpose of this study a key length of 256 bits is chosen to measure the bandwidth used to rekey. A 256 bit key is a standard key length in the popular Advanced Encryption Standard (AES). Although 128 bit length key is more commonly implemented, the larger key length is chosen to model the case where extra security is needed. As a result, the size of the network layer packet used to distribute the group secret key on a rekey is 688 bits as shown below:

Packet Size = MAC Header + CRC + Encryption Key + IP Header

= 240 bits + 32 bits + 256 bits + 160 bits

= 688 bits

The average bandwidth used to rekey is calculated by summing all the rekeys for each UAV performed over the simulation period multiplied by the packet size and divided by the number of seconds in the simulation. This gives the average bits per second (bps). This calculation assumes that each UAV is rekeyed directly by the cluster leader or Global Hawk, in the case of the baseline architecture, (the packets are not transmitted through intermediary UAVs). Also, this calculation only takes the packet

with the encryption key into account (management or acknowledgement packets are not used in the calculation because they depend the specific protocols used). This calculation also assumes a pair-wise rekey between the cluster leader (or Global Hawk) and each UAV, which results in one separate message for each UAV (n messages). An alternative to pair-wise rekeying and sending n messages is to multicast a single message containing n copies of the new group key each encrypted with a different members pair-wise key. This alternative results in only one message sent, however the message is of size O(n). Again, any rekeying scheme can be adapted depending on the needs of the application.

Appendix B. Battery Consumed to Rekey Calculations

This section outlines the calculations used to measure the percentage of battery consumed to rekey (*battery consumed*) and the necessary assumptions. The same assumptions used to calculate *average bandwidth*, as outlined in the Appendix G, are also applied to calculate *battery consumed*. In addition, assumptions about the battery and radio are necessary.

The representative battery chosen for the simulations is the Thunder Power Lithium Poly battery, which has a usable voltage range from 14 to 16.7 V, and a 4200 mA-hr capacity [Gru07]. This battery is currently being used to power UAVs for swarming applications [Gru07]. The representative radio chosen for the simulations is the Ubiquiti Networks SuperRange9 radio, which is also currently being used in conjunction with the selected battery in UAV research [Gru07]. The SuperRange9 is a 900 MHz wireless radio, which features up to 700 mW of output power, -88 dBm of receive sensitivity performance (for the 11Mbps data rate), and has proven non-line-of sight distances over 20km [Ubi07]. The current draw to transmit and receive are 1200 mA and 500 mA respectively [Ubi07]. The range and capabilities of the selected radio and battery make the assumed communication ranges for the three architectures viable.

With the battery and radio selected, there is enough information to calculate *battery consumed*. First, the energy consumed to rekey is found, which consists of the energy consumed to transmit the rekey packet and the energy consumed to receive the rekey packet. The equations used to calculate the energy consumed to receive and transmit are shown in Equation B.1 and Equation B.2 respectively [Jor06].

$$E_{Rx} = \frac{b_R \times d_R}{\left(3600\frac{s}{hr}\right)r} = \frac{688 \times 500mA}{\left(3600\frac{s}{hr}\right)11Mbps} = 0.0000087mA - hr \qquad \text{(B.1)}$$

$$E_{Tx} = \frac{b_T \times d_T}{\left(3600\frac{s}{hr}\right)r} = \frac{688 \times 1200mA}{\left(3600\frac{s}{hr}\right)11Mbps} = 0.0000208mA - hr \qquad \text{(B.2)}$$

The symbols used in the equations are defined in Table 12.

Table 12. Energy Consumption Symbols

E_{Rx}	Energy Consumed from Receiving (mA-hr)
E_{Tx}	Energy Consumed from Transmitting (mA-hr)
b_T	Bits Transmitted
b_R	Bits Received
d_T	Current Draw from Transmitting (mA-hr)
d_R	Current Draw from Transmitting (mA-hr)
r	Data Rate (bits/second)

The bits transmitted and received are the number of bits in the rekey packet (688). The

current draw from transmitting and receiving are taken from the radio's datasheet and the

data rate is assumed to be 11 Mbps, which is the maximum data rate for IEEE 802.11b.

Then, the results of Equations B.1 and B.2 are divided by the battery capacity to get a

percentage of battery consumed to receive a rekey packet and transmit a rekey packet.

% of Battery consumed to receive one key: $\frac{0.0000087mA - hr}{4200mA} \times 100 = 0.000000207\%$

% of Battery consumed to transmit one key: $\frac{0.00000208mA - hr}{4200mA} \times 100 = 0.000000496\%$

These equations are used in the simulation to calculate an overall average percentage of

battery consumed by a UAV to rekey during the simulation period.

Appendix C. Rekey Time Interval Considerations

In this study, the rekey time interval is defined to be the length of time between system checks for events, such as a join or departure, which create the need for a rekey operation. The shorter the rekey time interval, the more secure the system will be. However, short rekey time intervals also increase the security overhead and overall traffic in the system. For example, if the rekey time interval is set to 60 seconds, a UAV could continue to receive up to 60 seconds of multicast traffic after it has been evicted (depending the point during the rekey interval the eviction occurred). If the rekey interval is set to one tenth of a second, the amount of multicast traffic received after an eviction would be significantly less (600 times less).

The rekey interval also affects the amount traffic in the system associated with rekeying. Using the same example, during the 60 seconds of elapsed time their may have been 10 events that would have triggered several rekey operations, but because the interval is 60 seconds, only one rekey operation would occur at the end of the interval. With the rekey interval at one tenth of a second all 10 rekey operations would trigger a rekey operation, assuming they were separated by at least one tenth of a second, creating 10 times the traffic in the system.

Therefore, the rekey time interval needs to balance both security requirements and system constraints such as bandwidth and processing power. For the highest security levels the rekey time interval would be set to zero, meaning virtual real-time rekey operations would occur.

Appendix D. Additional Total Keys Distributed Plots for Scenario 1

95% CI for the Mean
SwarmSize = 40, Mobility = 25%
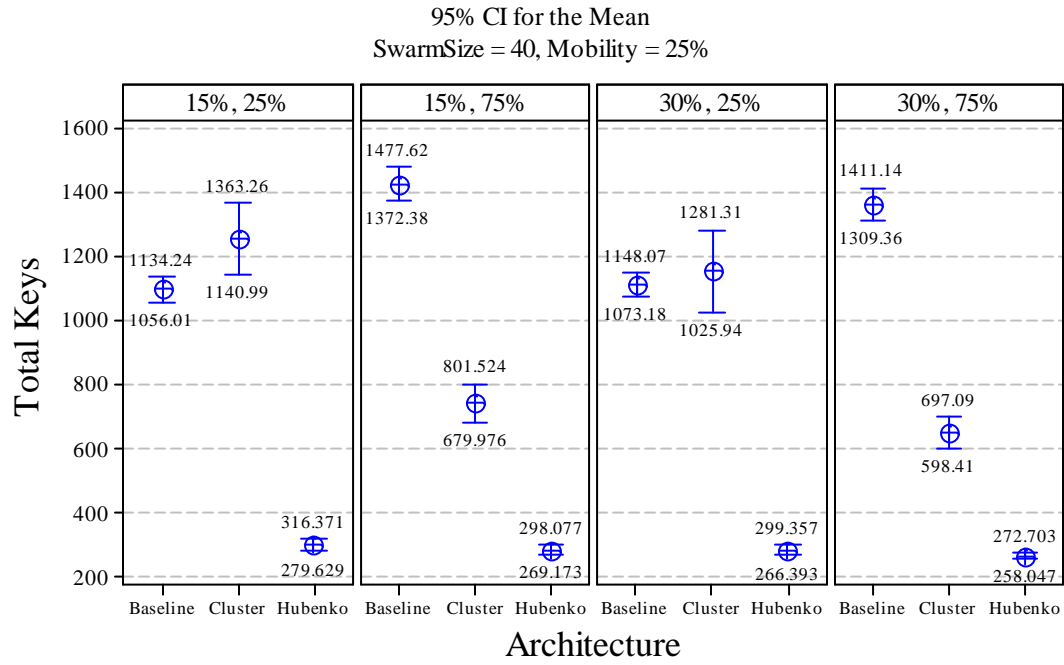
Panel variables: Group Join Rate, Group Departure Rate

Figure 29. Total Keys versus Architecture with Swarm Size of 40 and 25% Mobility
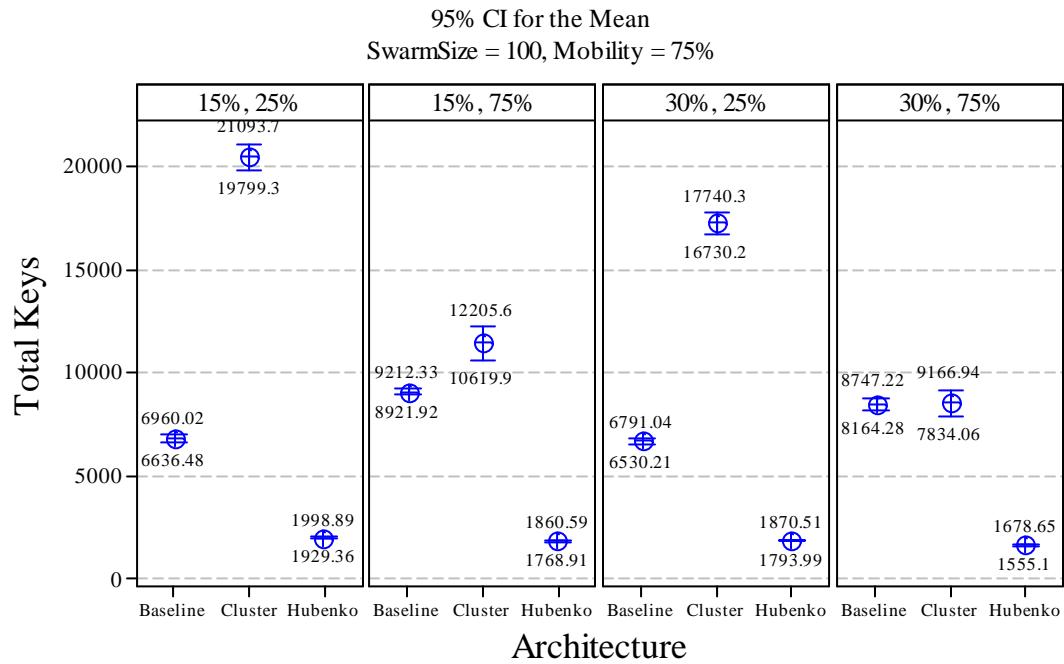
95% CI for the Mean
SwarmSize = 40, Mobility = 75%
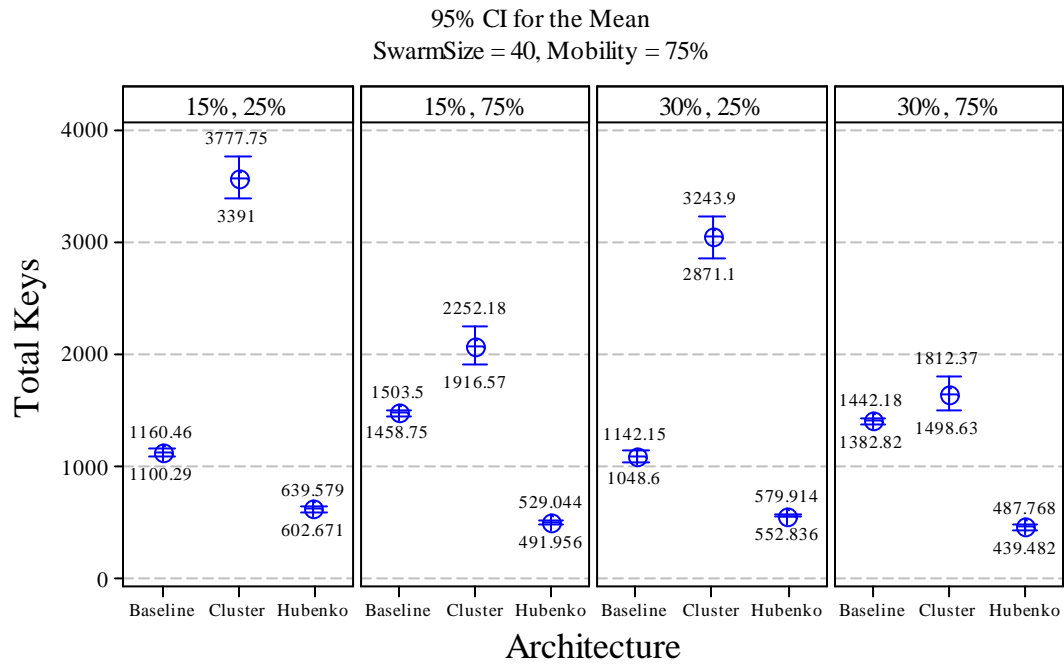
Panel variables: Group Join Rate, Group Departure Rate

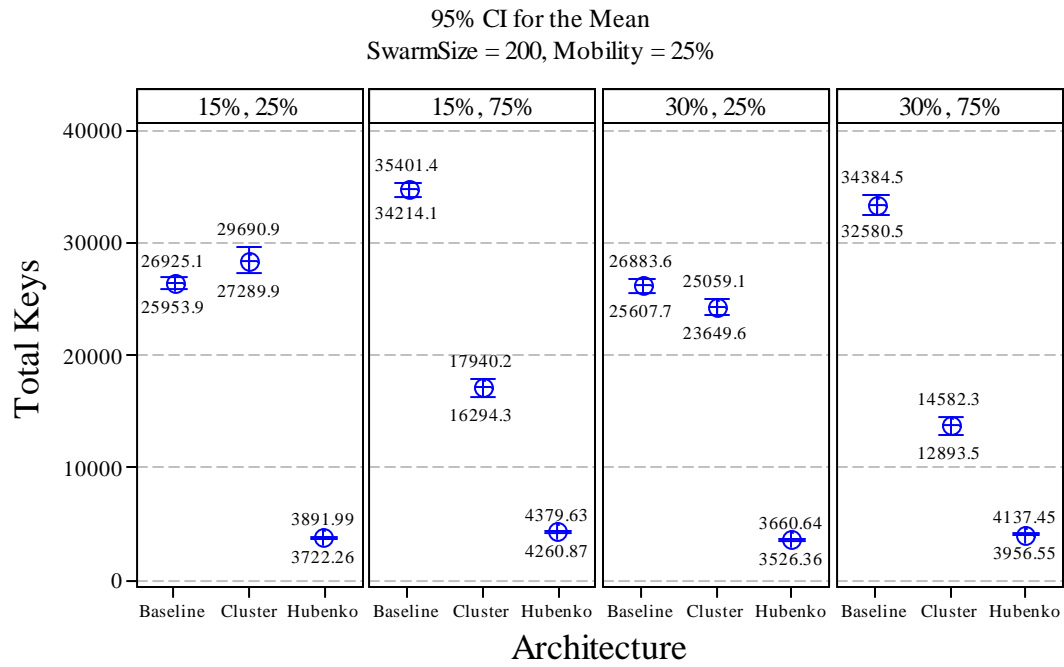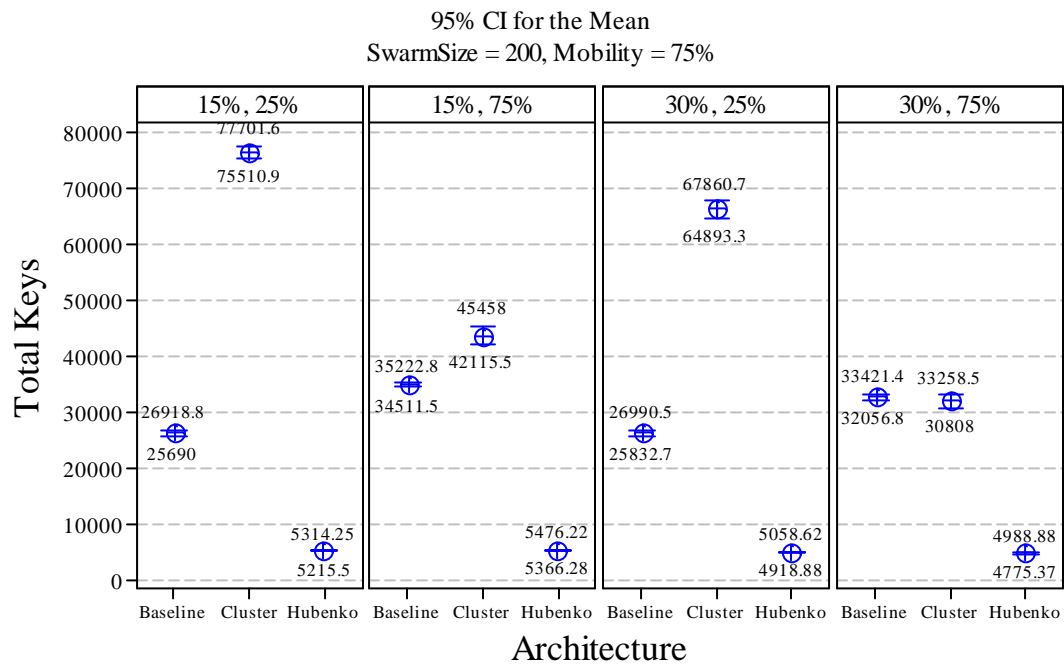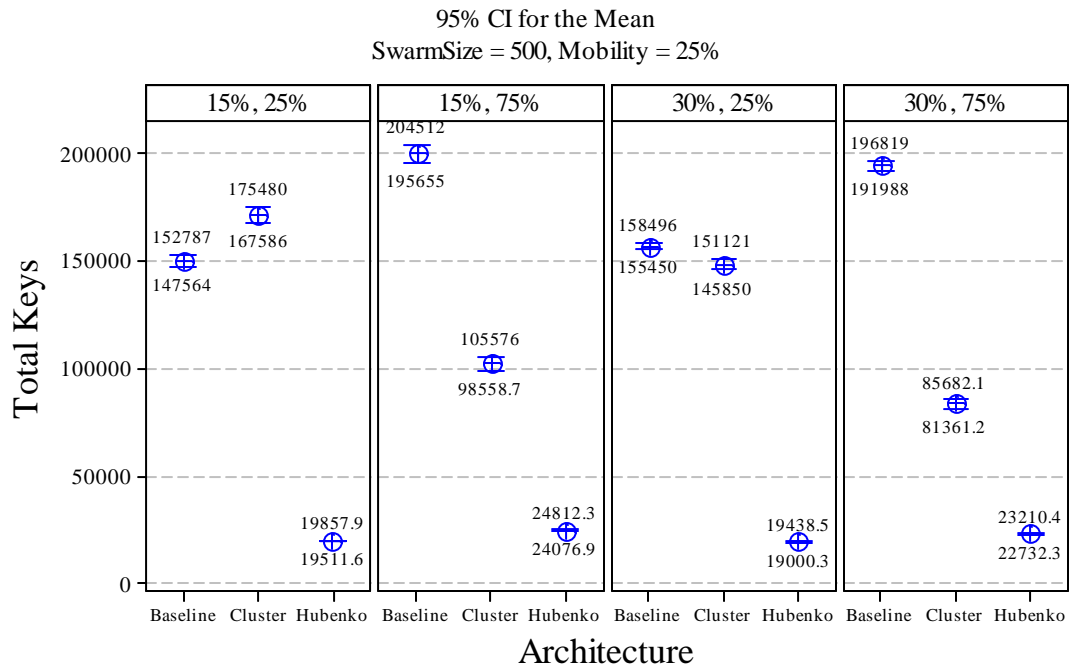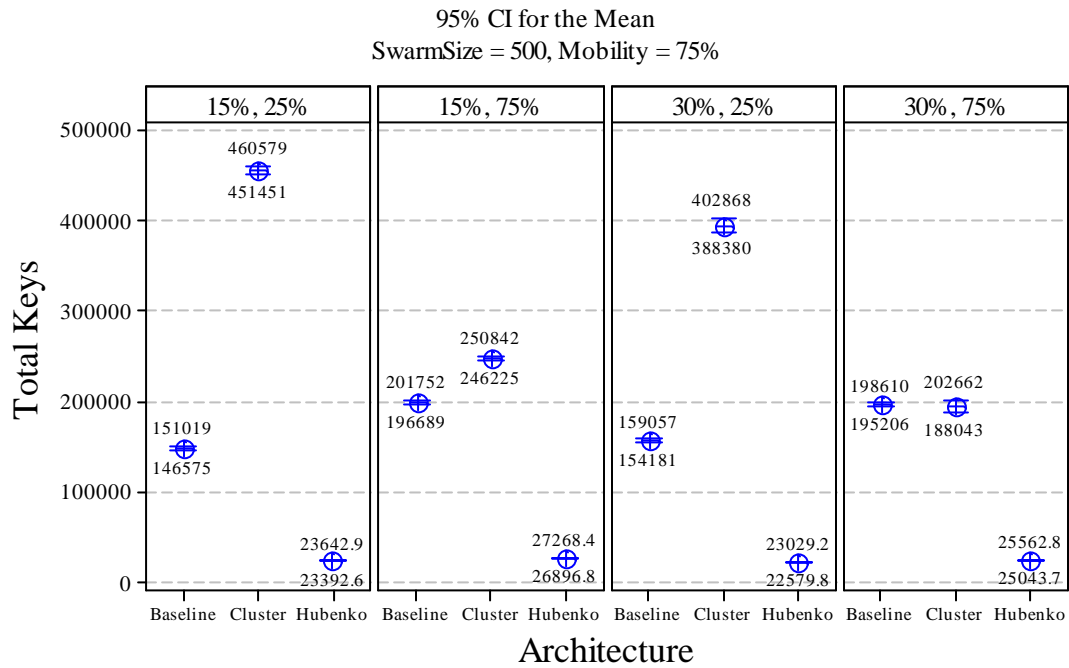Figure 30. Total Keys versus Architecture with Swarm Size of 40 and 75% Mobility

95% CI for the Mean
SwarmSize = 100, Mobility = 25%
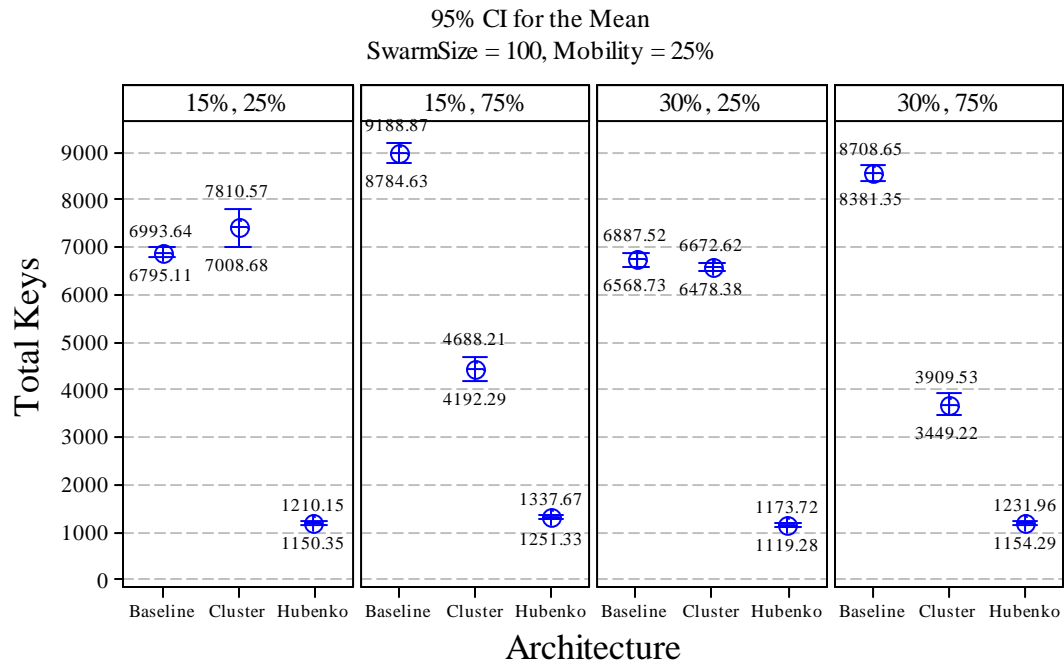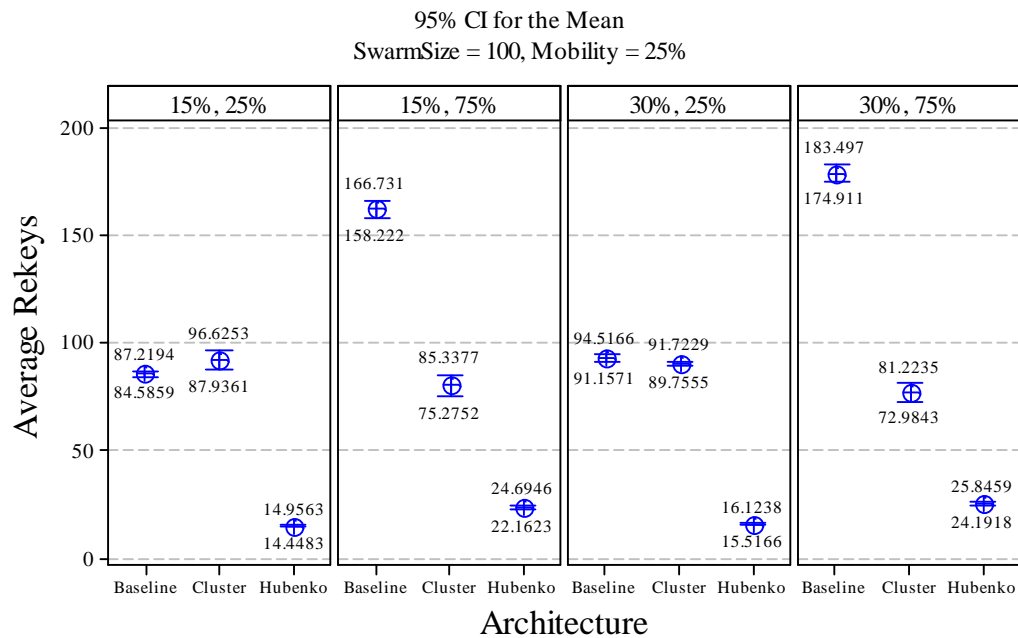
Panel variables: Group Join Rate, Group Departure Rate

Figure 31. Total Keys versus Architecture with Swarm Size of 100 and 25% Mobility

95% CI for the Mean
SwarmSize = 100, Mobility = 75%
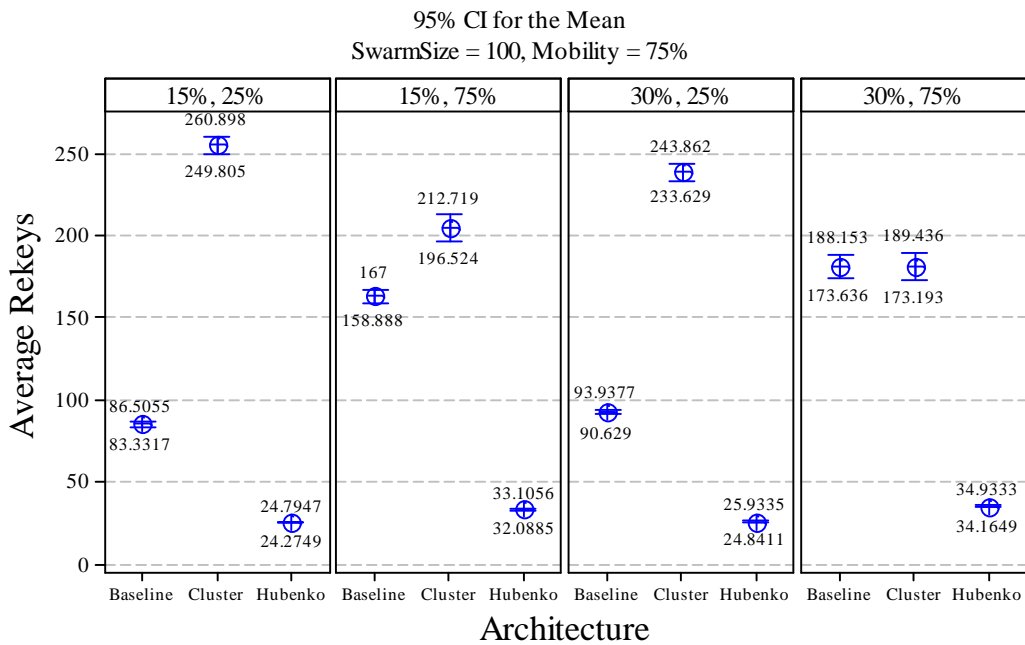
Panel variables: Group Join Rate, Group Departure Rate

Figure 32. Total Keys versus Architecture with Swarm Size of 100 and 75% Mobility

Figure 33. Total Keys versus Architecture with Swarm Size of 200 and 25% Mobility

Figure 34. Total Keys versus Architecture with Swarm Size of 200 and 75% Mobility

95% CI for the Mean
SwarmSize = 500, Mobility = 25%

Panel variables: Group Join Rate, Group Departure Rate

Figure 35. Total Keys versus Architecture with Swarm Size of 500 and 25% Mobility

95% CI for the Mean
SwarmSize = 500, Mobility = 75%

Panel variables: Group Join Rate, Group Departure Rate

Figure 36. Total Keys versus Architecture with Swarm Size of 500 and 75% Mobility

Appendix E. Additional Average Rekey Plots for Scenario 1
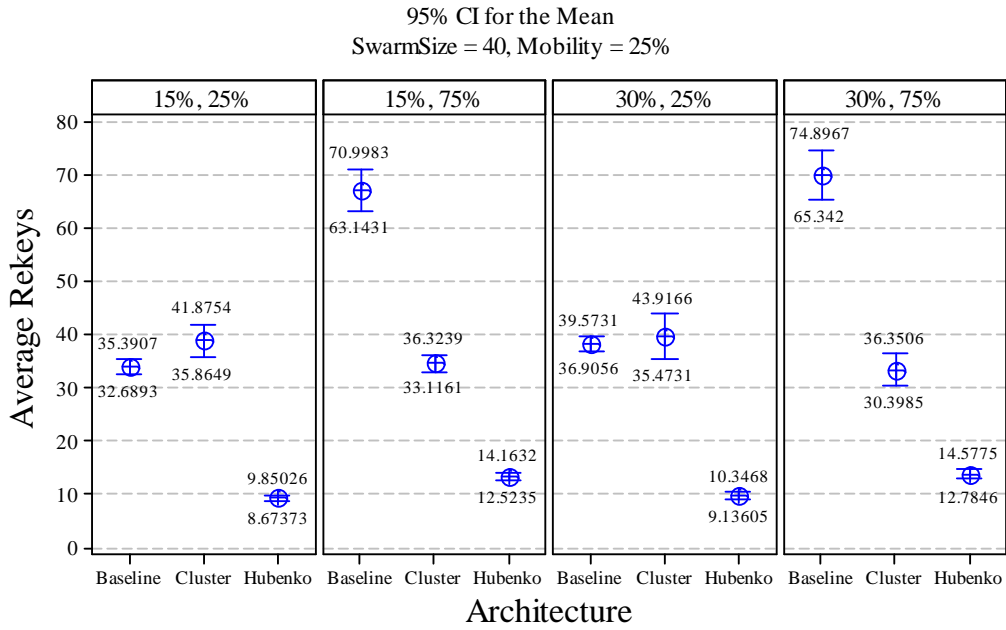
Figure 37. Average Rekeys versus Architecture with Swarm Size of 40 and 25% Mobility
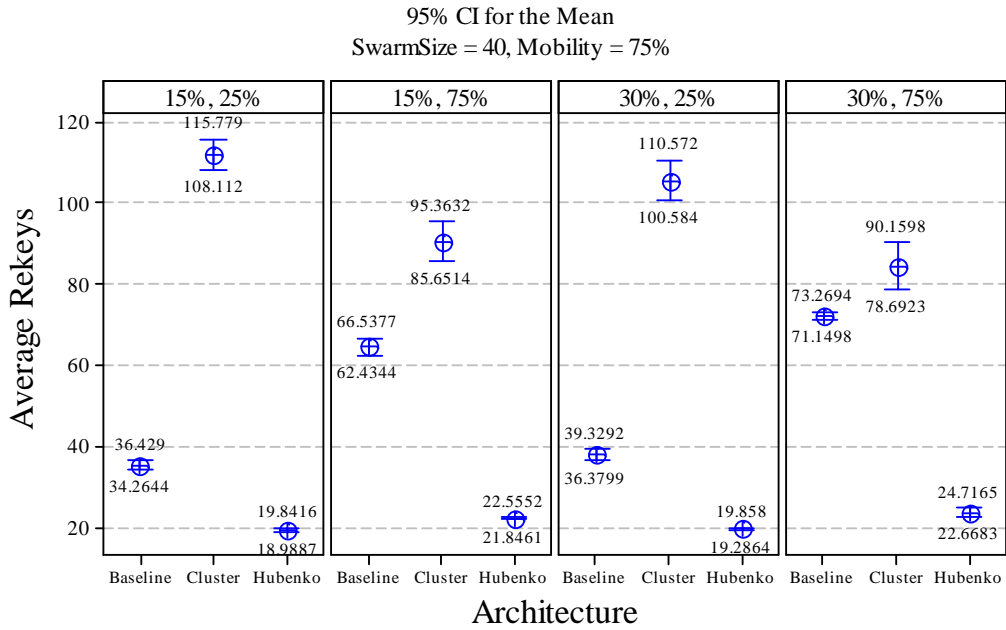
Figure 38. Average Rekeys versus Architecture with Swarm Size of 40 and 75% Mobility
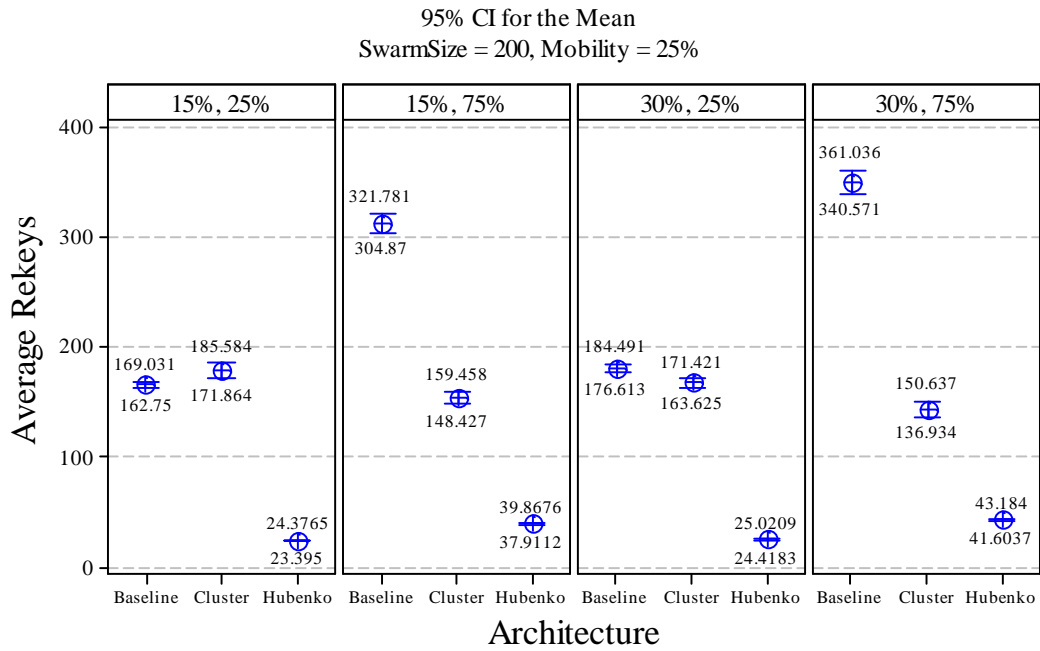
95% CI for the Mean
SwarmSize = 100, Mobility = 25%

Panel variables: Group Join Rate, Group Departure Rate

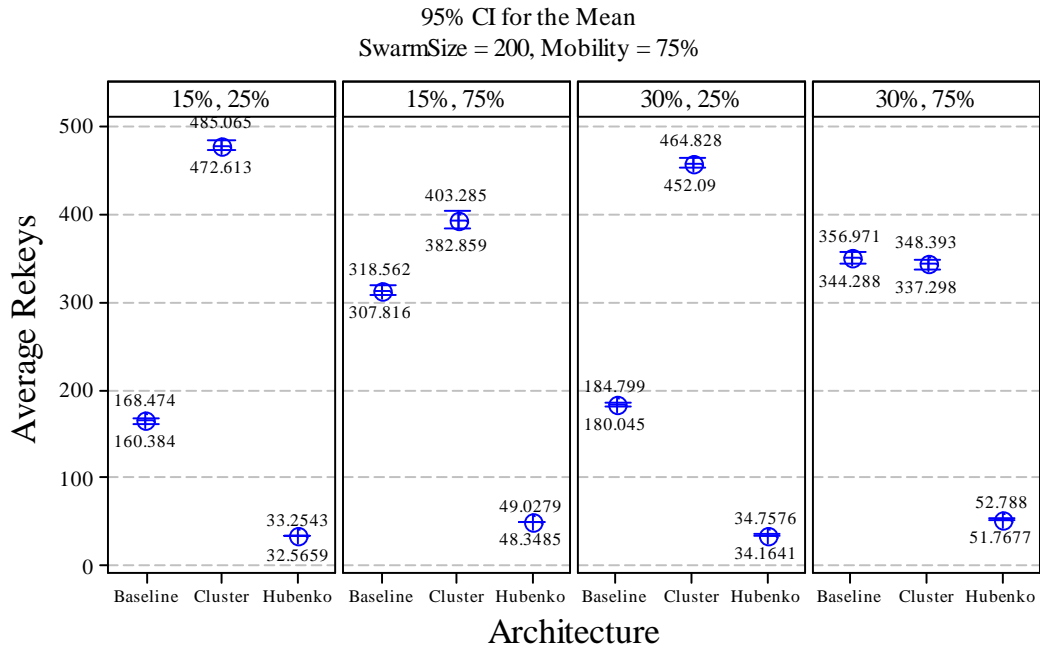Figure 39. Average Rekeys versus Architecture with Swarm Size of 100 and 25% Mobility

95% CI for the Mean
SwarmSize = 100, Mobility = 75%

Panel variables: Group Join Rate, Group Departure Rate

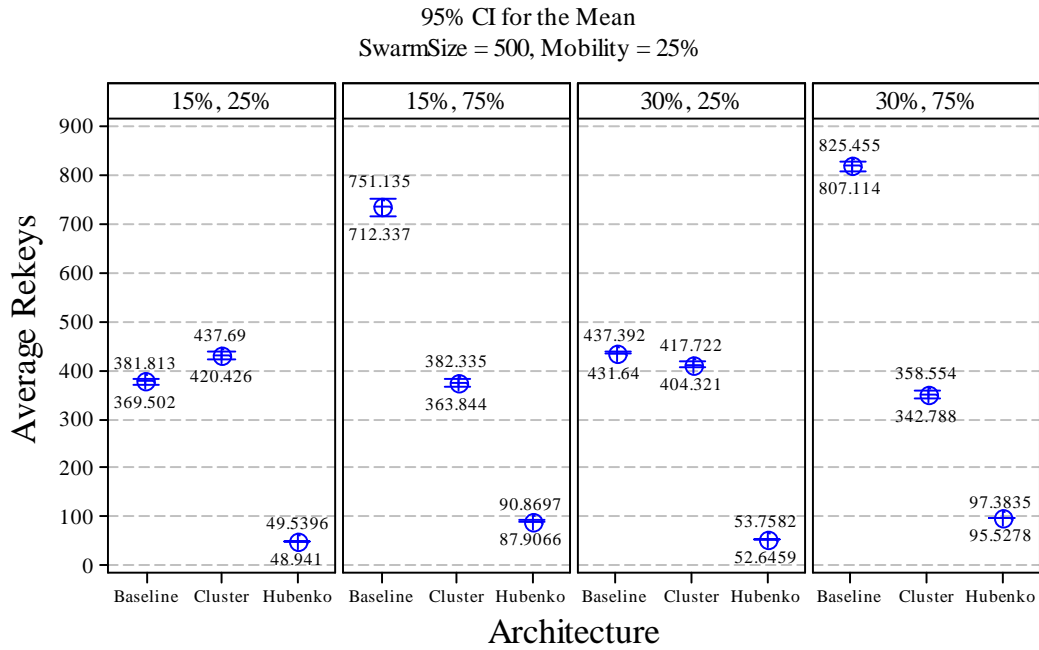Figure 40. Average Rekeys versus Architecture with Swarm Size of 100 and 75% Mobility

95% CI for the Mean
SwarmSize = 200, Mobility = 25%

Panel variables: Group Join Rate, Group Departure Rate

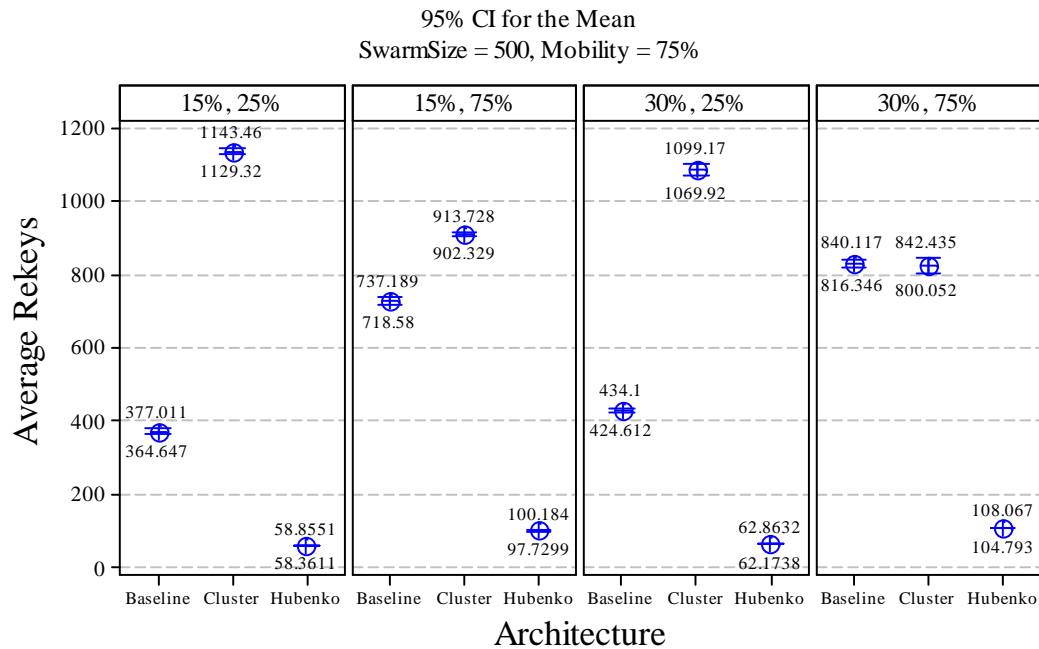Figure 41. Average Rekeys versus Architecture with Swarm Size of 200 and 25% Mobility

95% CI for the Mean
SwarmSize = 200, Mobility = 75%

Panel variables: Group Join Rate, Group Departure Rate

Figure 42. Average Rekeys versus Architecture with Swarm Size of 200 and 75% Mobility

95% CI for the Mean
SwarmSize = 500, Mobility = 25%

Panel variables: Group Join Rate, Group Departure Rate

Figure 43. Average Rekeys versus Architecture with Swarm Size of 500 and 25% Mobility

95% CI for the Mean
SwarmSize = 500, Mobility = 75%

Panel variables: Group Join Rate, Group Departure Rate

Figure 44. Average Rekeys versus Architecture with Swarm Size of 500 and 75% Mobility

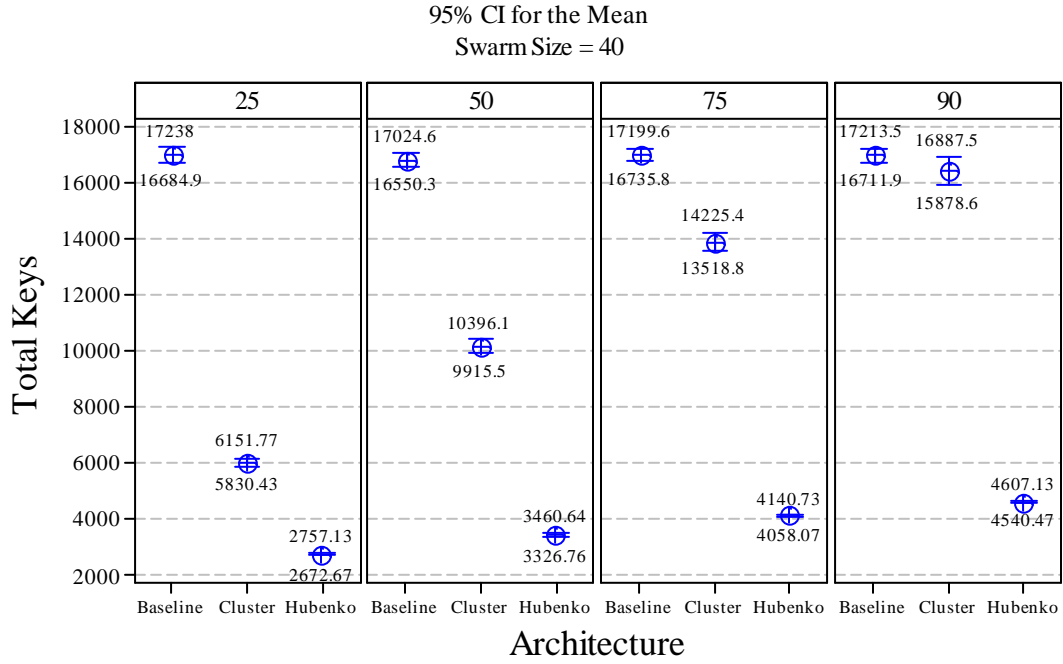**Appendix F. Additional Total Keys Distributed Plots for Scenario 2**

95% CI for the Mean
Swarm Size = 40

Panel variable: Percent Mobile

Figure 45. Total Keys versus Architecture with Swarm Size of 40

95% CI for the Mean
Swarm Size = 100

Panel variable: Percent Mobile
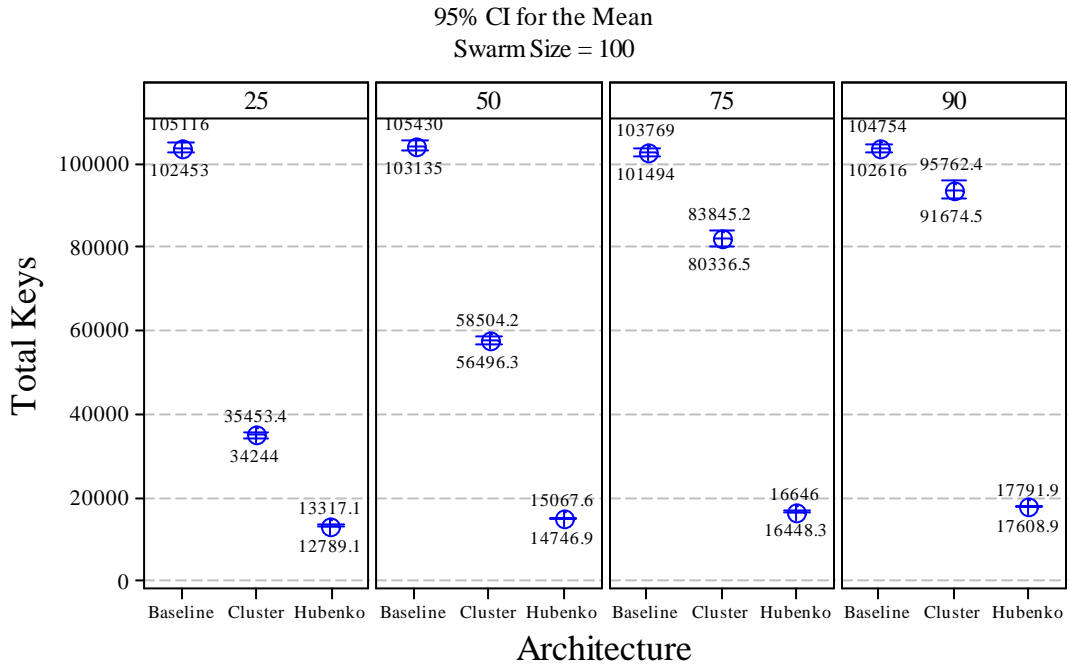
Figure 46. Total Keys versus Architecture with Swarm Size of 100

95% CI for the Mean
Swarm Size = 200
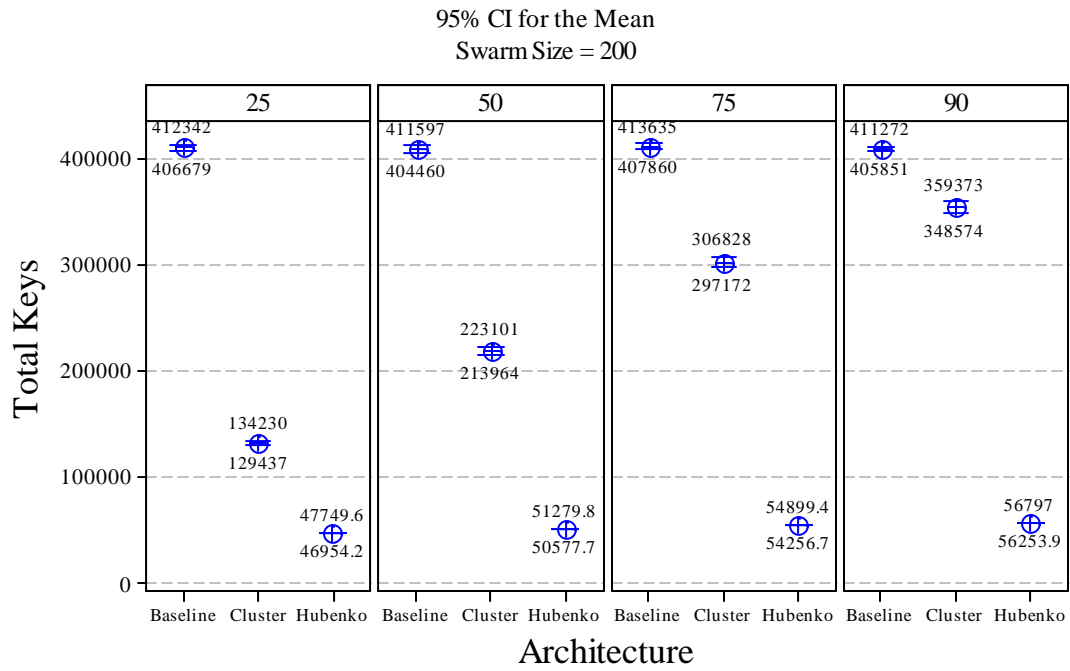
Figure 47. Total Keys versus Architecture with Swarm Size of 200

95% CI for the Mean
Swarm Size = 500
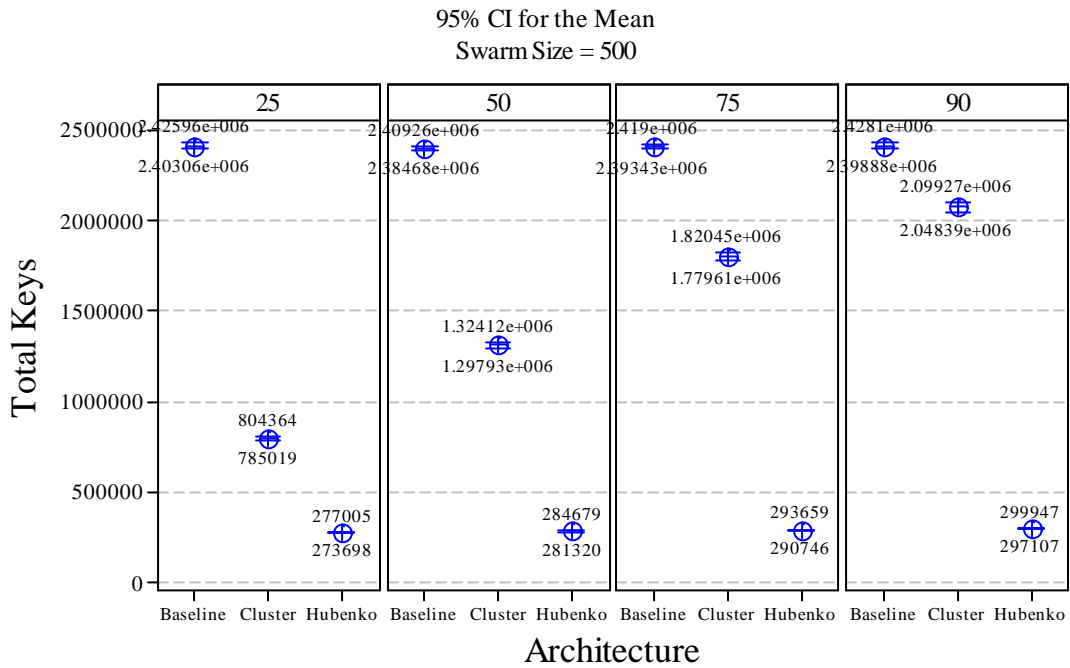
Figure 48. Total Keys versus Architecture with Swarm Size of 500

Swarm Size = 1000

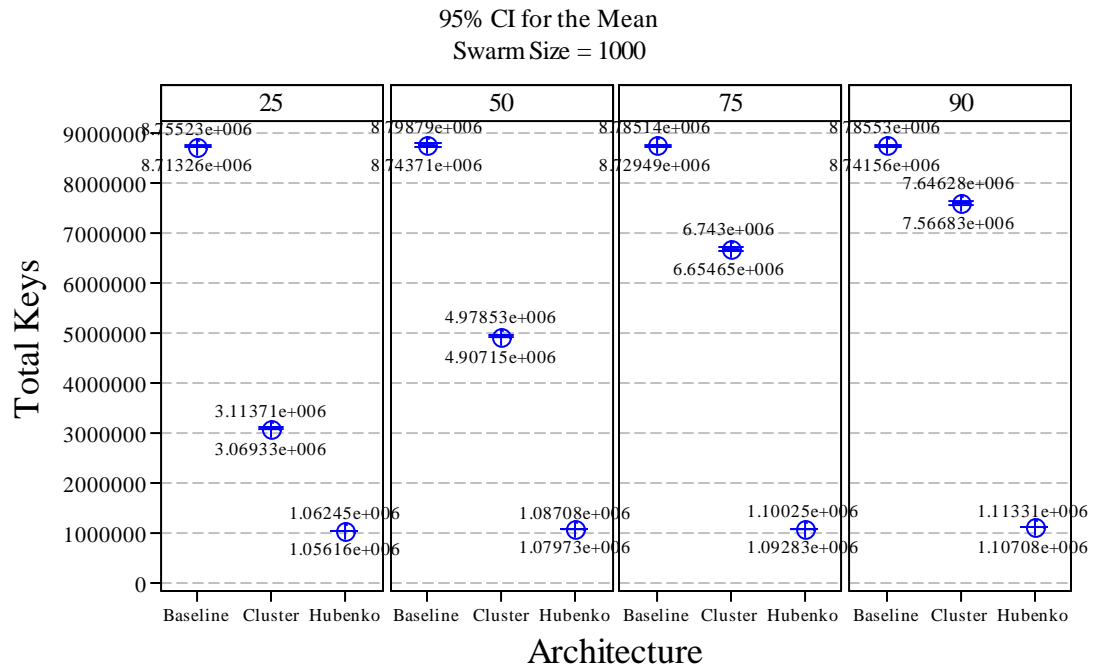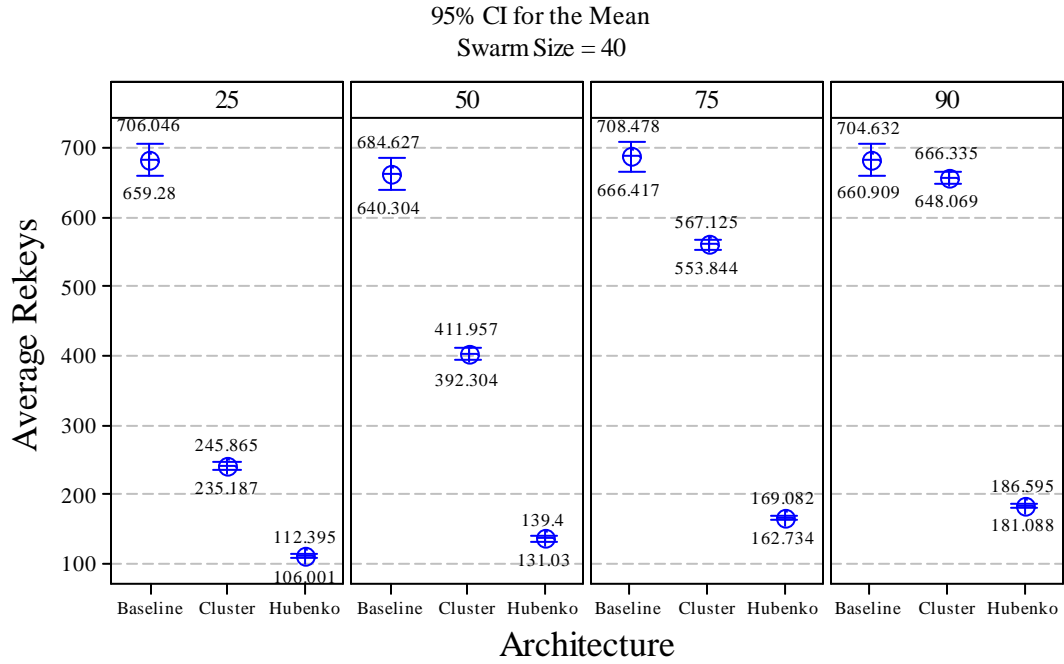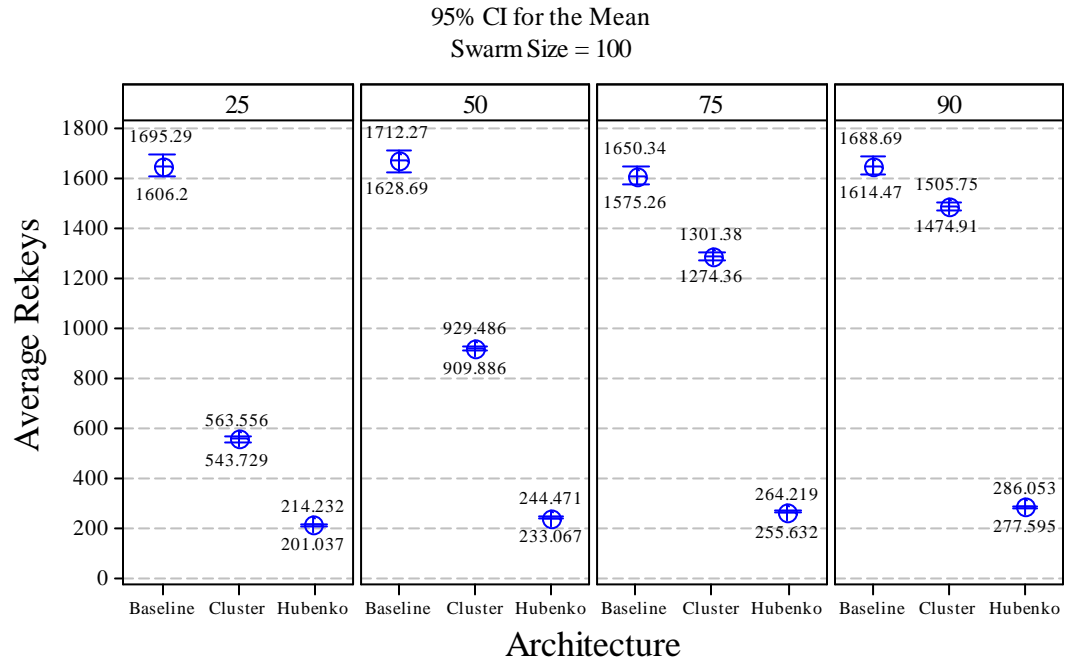Figure 49. Total Keys versus Architecture with Swarm Size of 100

Appendix G. Additional Average Rekey Plots for Scenario 2

95% CI for the Mean
Swarm Size = 40

Panel variable: Percent Mobile

Figure 50. Average Rekeys versus Architecture with Swarm Size of 40

95% CI for the Mean
Swarm Size = 100

Panel variable: Percent Mobile

Figure 51. Average Rekeys versus Architecture with Swarm Size of 100

95% CI for the Mean
Swarm Size = 200
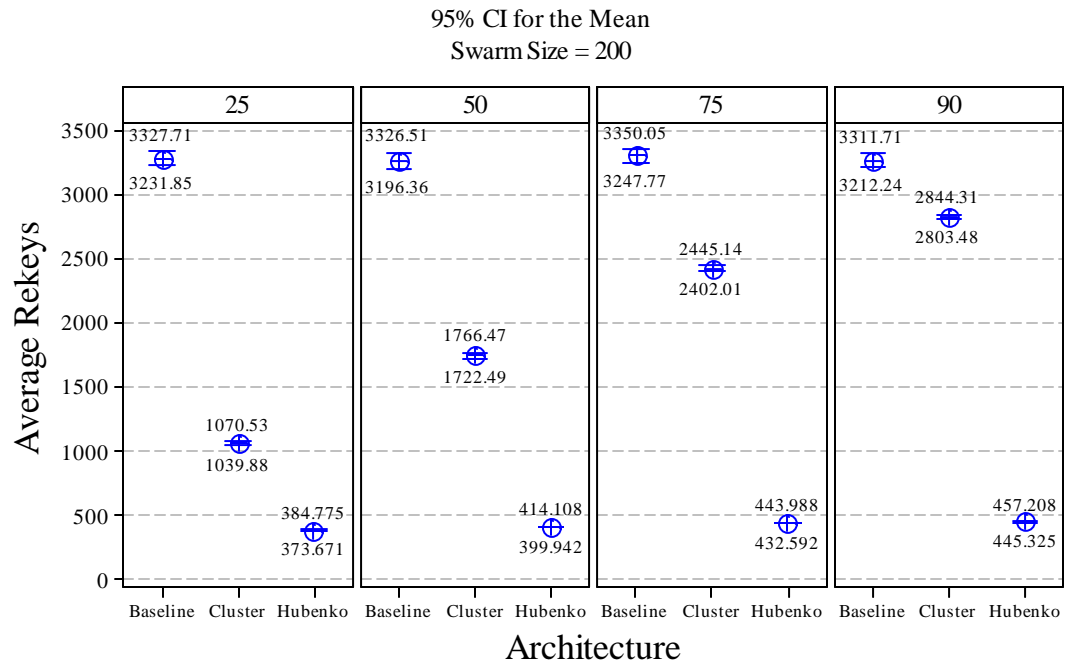
Figure 52. Average Rekeys versus Architecture with Swarm Size of 200

Panel variable: Percent Mobile

95% CI for the Mean
Swarm Size = 500
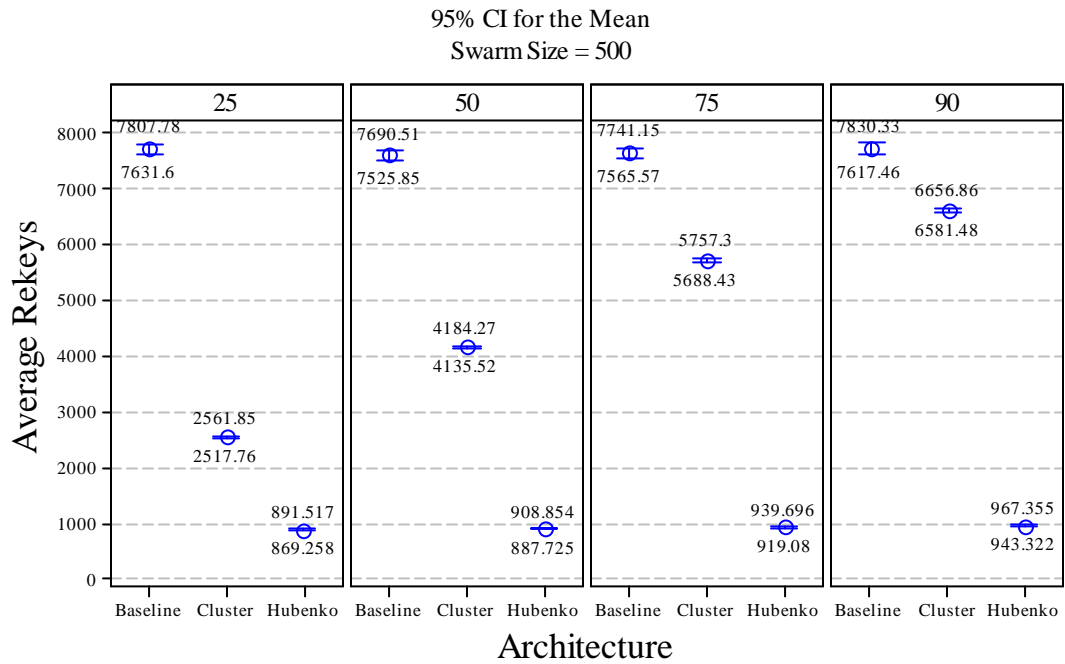
Figure 53. Average Rekeys versus Architecture with Swarm Size of 500

Panel variable: Percent Mobile

95% CI for the Mean
Swarm Size = 1000
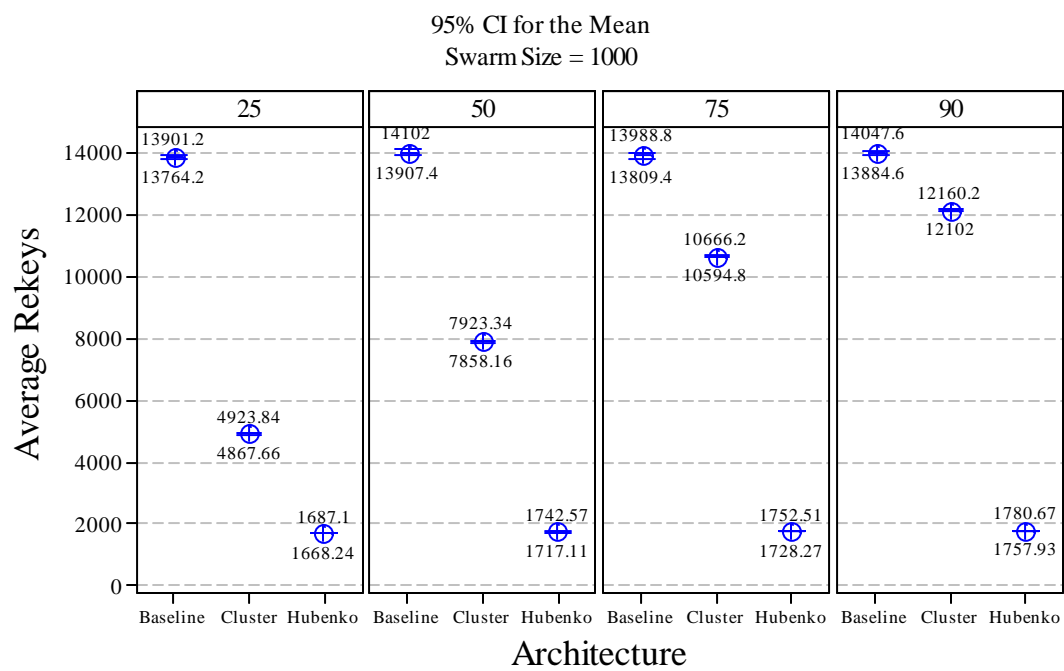
Figure 54. Average Rekeys versus Architecture with Swarm Size of 1000

Panel variable: Percent Mobile

Appendix H. Additional Average Bandwidth Plots for Scenario 2

95% CI for the Mean
Swarm Size = 40
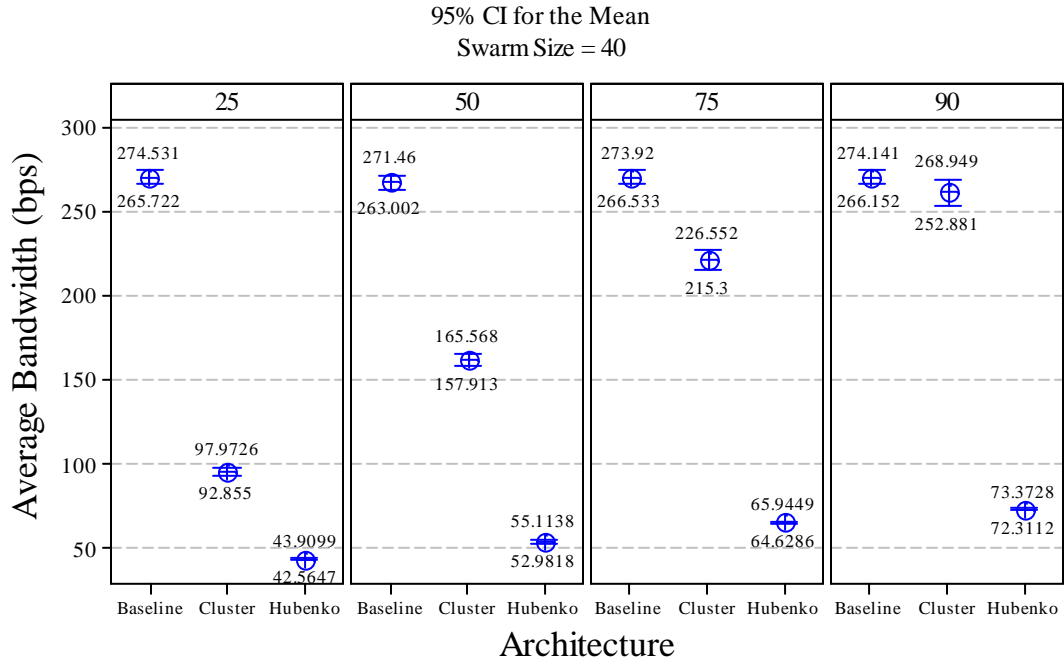
Panel variable: Percent Mobile

Figure 55. Average Bandwidth versus Architecture with Swarm Size of 40

95% CI for the Mean
Swarm Size = 100
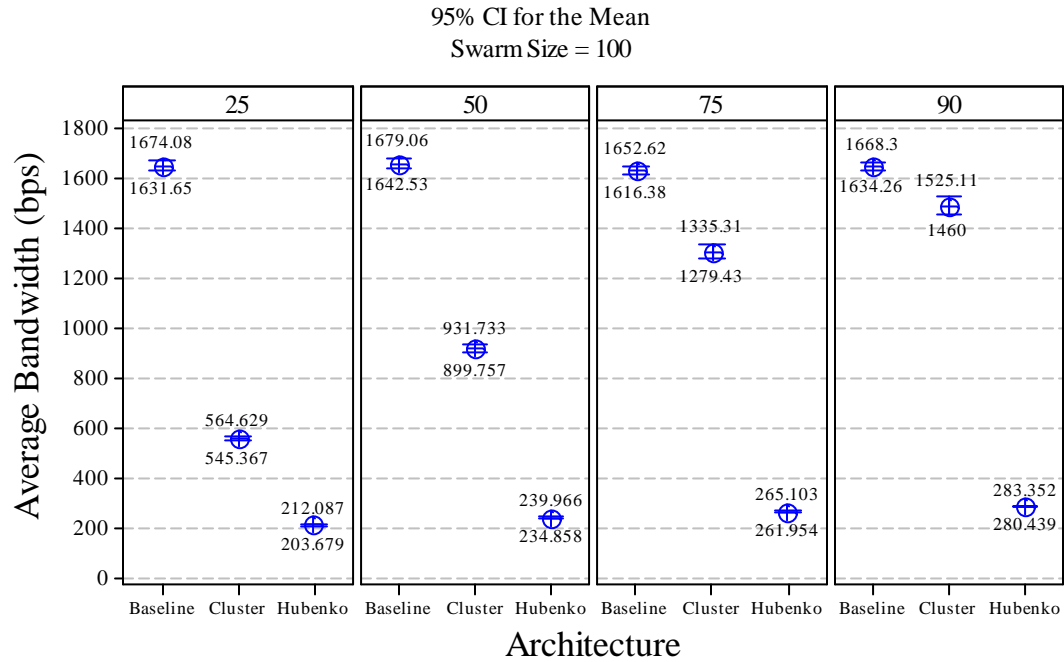
Panel variable: Percent Mobile

Figure 56. Average Bandwidth versus Architecture with Swarm Size of 100

95% CI for the Mean
Swarm Size = 200
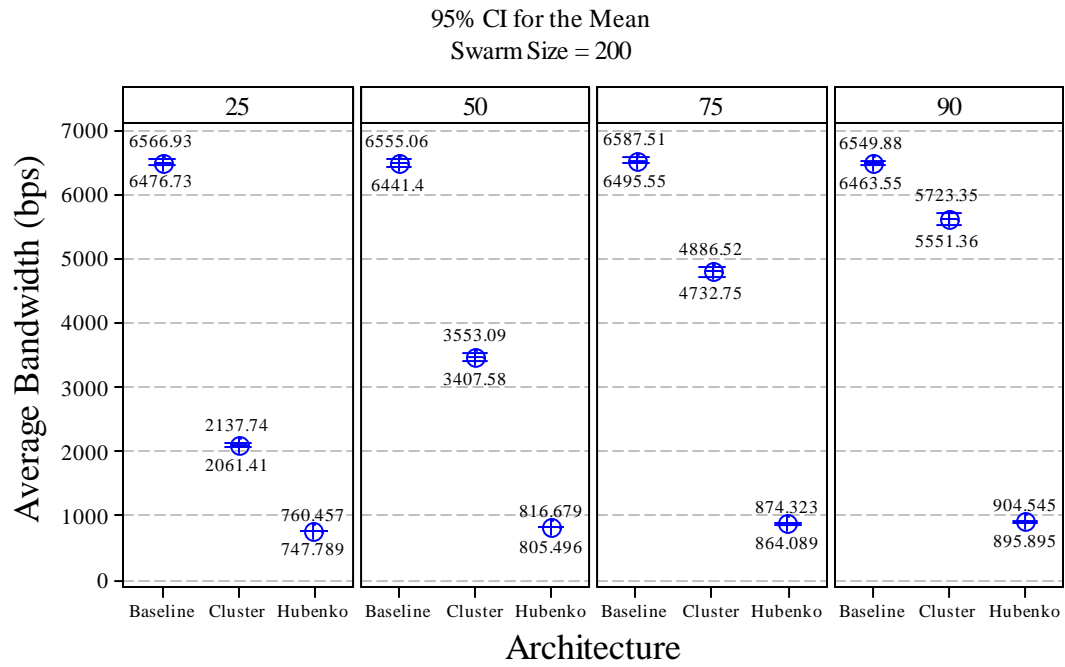
Panel variable: Percent Mobile

Figure 57. Average Bandwidth versus Architecture with Swarm Size of 200

95% CI for the Mean
Swarm Size = 500
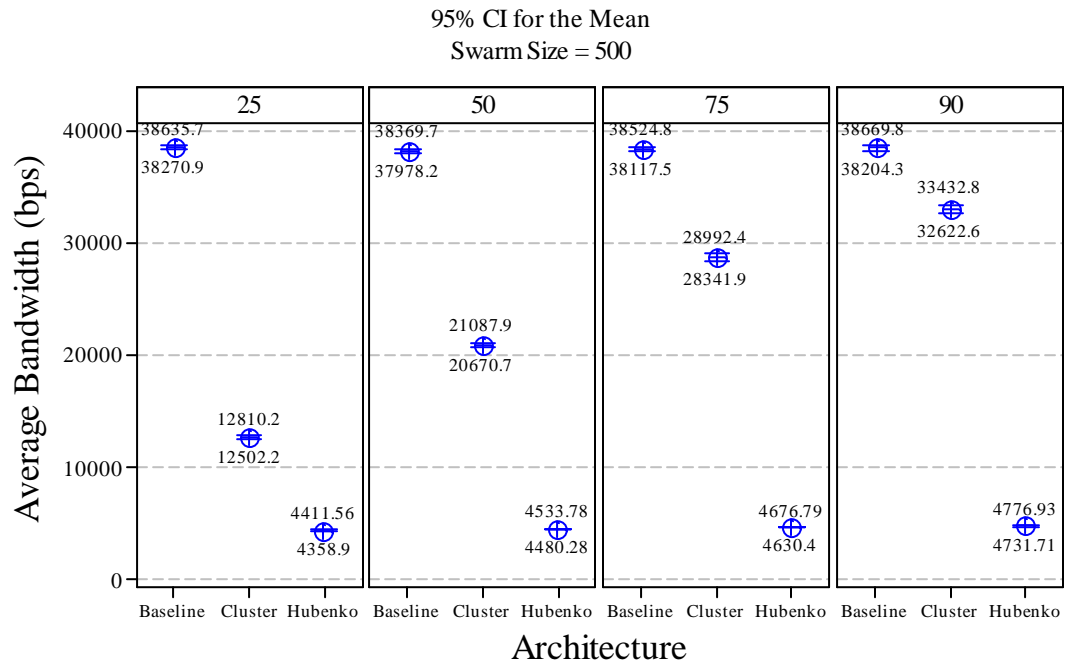
Panel variable: Percent Mobile

Figure 58. Average Bandwidth versus Architecture with Swarm Size of 500

95% CI for the Mean
Swarm Size = 1000
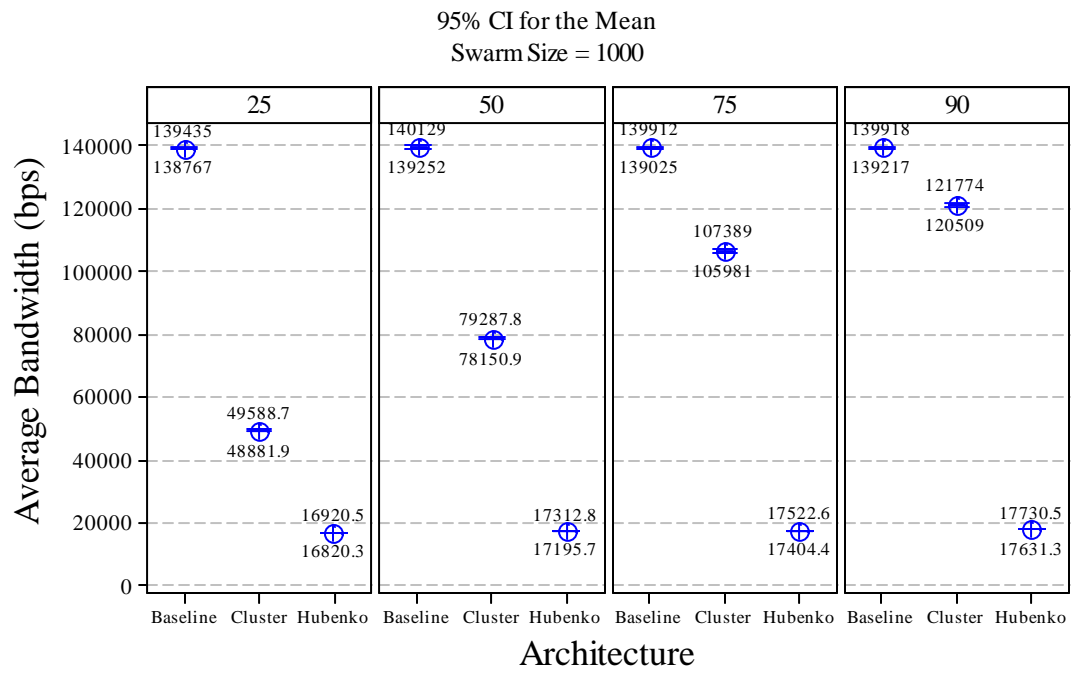
Figure 59. Average Bandwidth versus Architecture with Swarm Size of 1000

Panel variable: Percent Mobile

Appendix I. Additional Battery Consumed Plots for Scenario 2

95% CI for the Mean
Swarm Size = 40
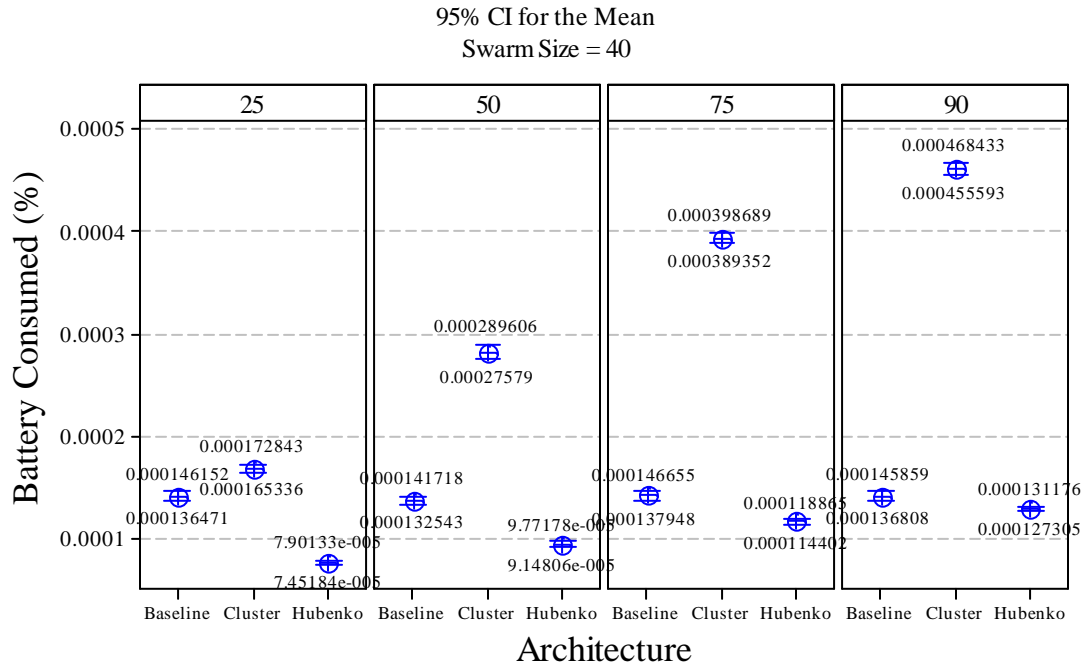
Panel variable: Percent Mobile

Figure 60. Battery Consumed versus Architecture with Swarm Size of 40

95% CI for the Mean
Swarm Size = 100
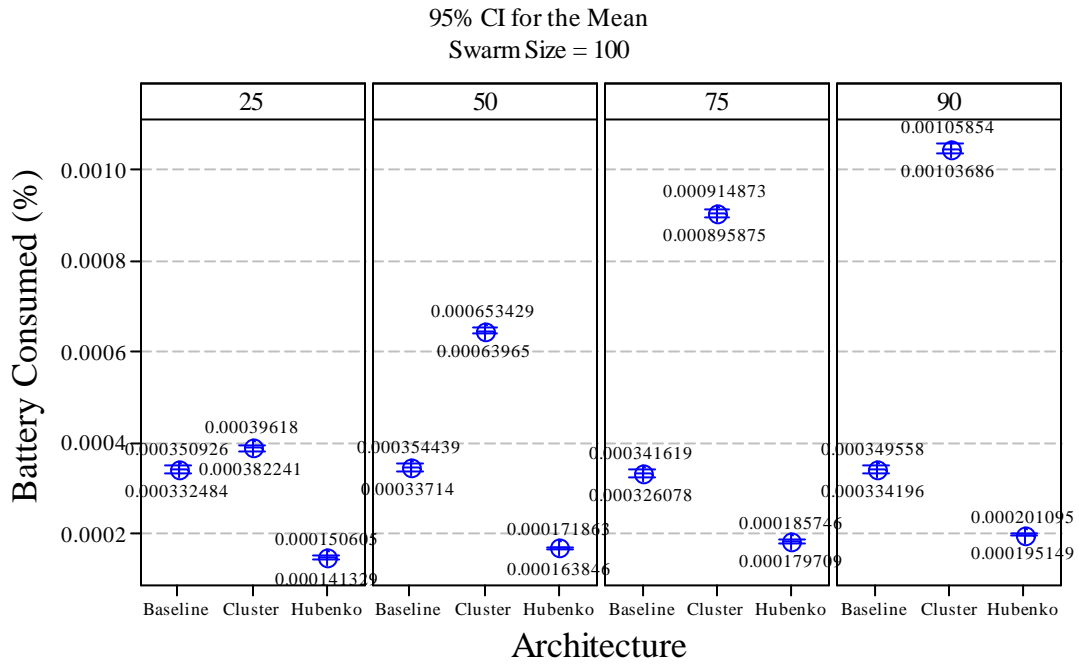
Panel variable: Percent Mobile

Figure 61. Battery Consumed versus Architecture with Swarm Size of 100

95% CI for the Mean
Swarm Size = 200
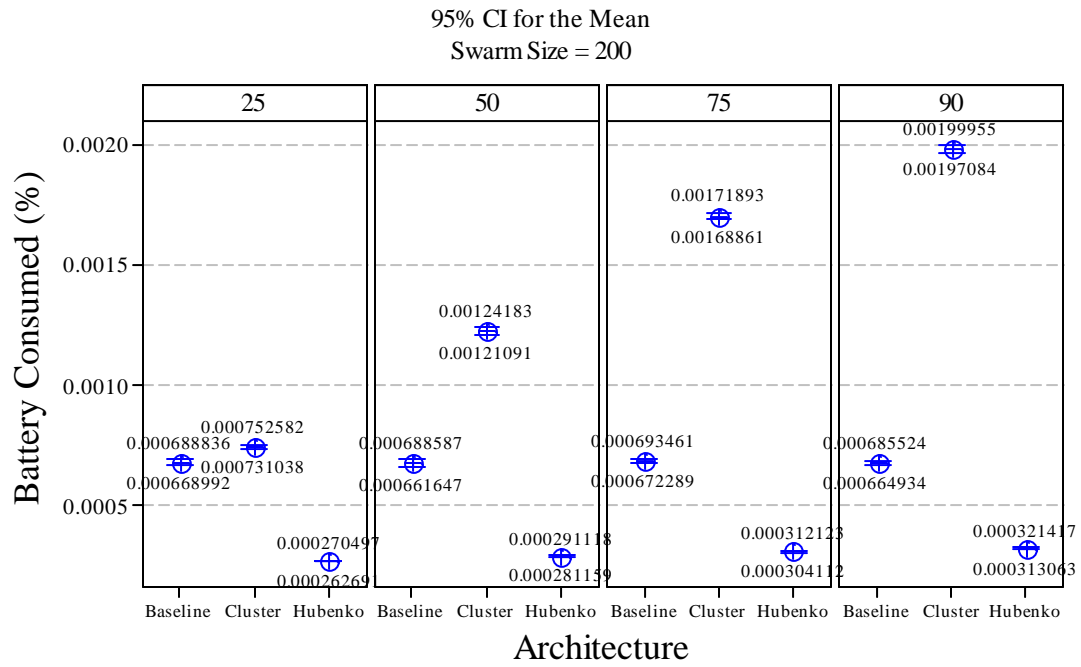
Panel variable: Percent Mobile

Figure 62. Battery Consumed versus Architecture with Swarm Size of 200

95% CI for the Mean
Swarm Size = 500
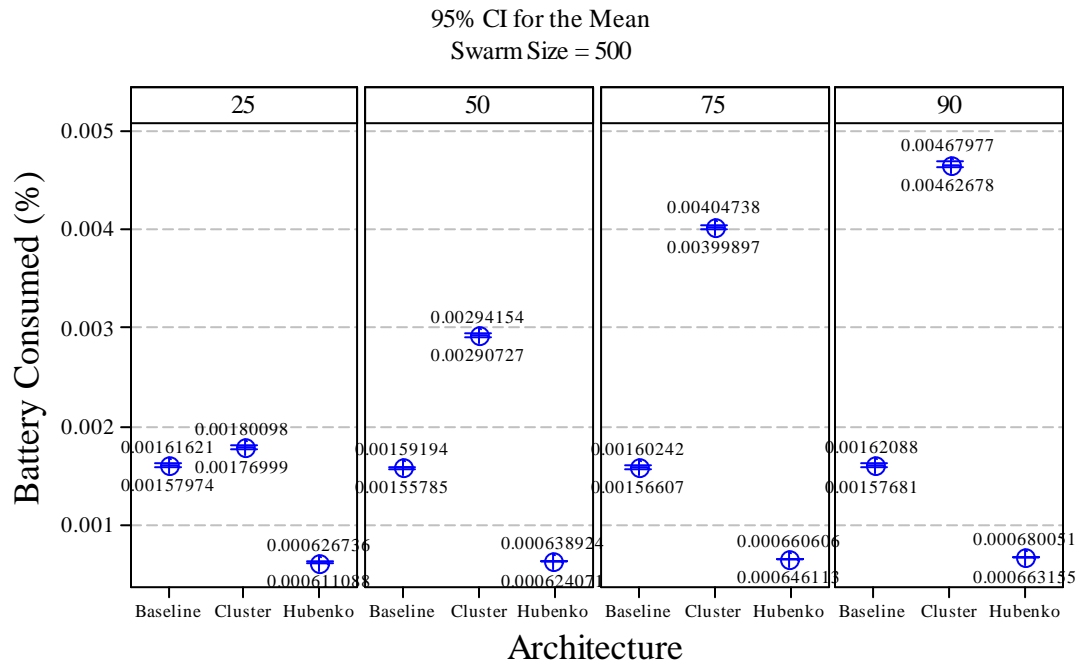
Panel variable: Percent Mobile

Figure 63. Battery Consumed versus Architecture with Swarm Size of 500

95% CI for the Mean
Swarm Size = 1000

Figure 64. Battery Consumed versus Architecture with Swarm Size of 1000

Panel variable: Percent Mobile

Bibliography

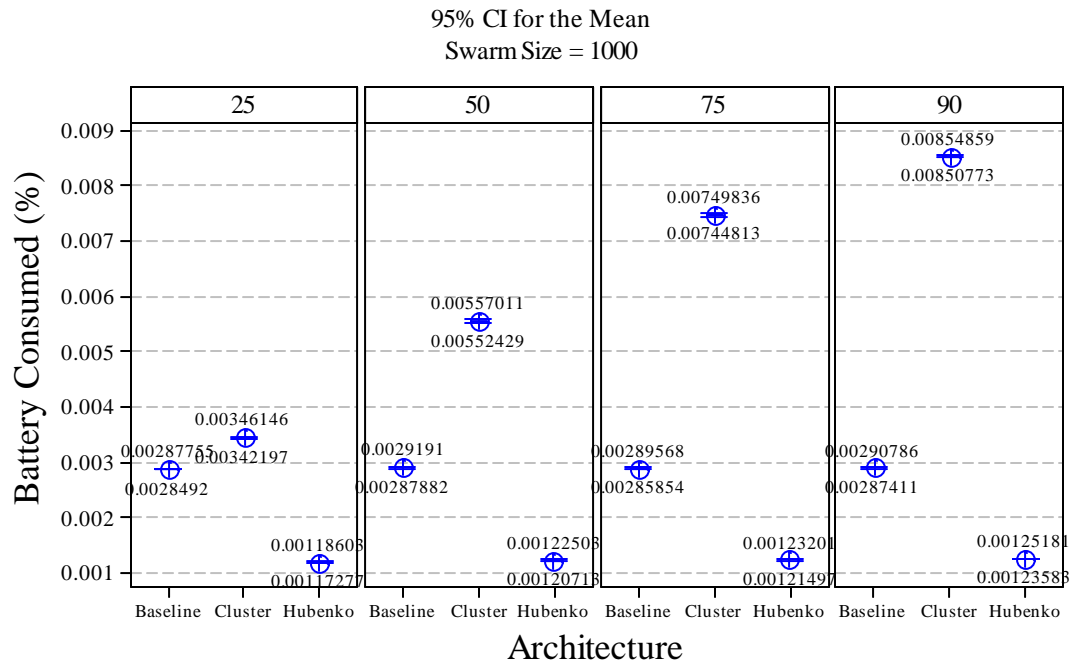[AdN05]   A. Adams, J. Nicholas, and W. Siadak, "Protocol Independent Multicast – Dense Mode (PIM-DM): Protocol Specification (Revised)," RFC 3973, IETF Network Working Group, 2005.

[AlH03]   D. Alberts, and R. Hayes, *Power to the Edge: Command and Control in the Information Age*. Washington, DC: DoD Command and Control Research Project, 2003. Available http://www.dodccrp.org/files/Alberts_Power.pdf.

[AuM06]   C. Augeri, K. Morris, and B. Mullins, "HARVEST: A framework and co-simulation for analyzing unmanned aerial vehicle swarms," *Military Communications Conference*, vol. 23-25 Oct 2006, pp. 1-7, 2006.

[AyS06]   W. Aye, and M. U. Siddiqi, "Key Management for Secure Multicast over IPv6 Wireless Networks," *EURASIP Journal on Wireless Communications and Networking,* vol. 2006, no. 61769, pp. 1-12, 2006.

[BaB02]   S. Banerjee and B. Bhattacharjee, "Scalable secure group communication over IP multicast," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1511-1527, 2002.

[Bis03]   M. Bishop, *Computer Security Art and Science*. Boston: Addison-Wesley, 2003.

[BoM98]   E. Bommaiah, A. McAuley, M. Liu, and R. Talpade "AMRoute: Adhoc Multicast Routing Protocol," Internet Draft, IETF, 1998.

[BrR02]   D. Bruschi, and E Rosti, "Secure Multicast in Wireless Networks of Mobile Hosts: Protocols and Issues," *Mobile Networks and Applications,* vol. 7, pp. 503-511, 2002.

[CaB02]   T. Camp, J. Boleng and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, pp. 483-502, 2002.

[Com99]   CompassRose (1999), "Introduction to Global Satellite Systems," *CompassRose International Publications,* Retrieved May 19, 2007 from http://www.compassroseintl.com/pubs/Intro_to_sats.html.

[DeG03]   C. De Morais Cordeiro, H. Gossain, and D. Agrawal, "Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions," *IEEE Network*, vol. 17, no. 1, pp. 52-29, 2003.

[DeU05]   Defense Update (2005), "Miniature Aerial Vehicles," Retrieved May 5, 2007 from http://www.defense-update.com/features/du-2-04/feature-mav.htm.

[DoD01]   Department of Defense (DoD). *Department of Defense Dictionary of Military and Associated Terms*. Joint Pub 1-02. Washington: HQ DoD, April 2001 (as amended through 22 March 2007). Available: http://www.dtic.mil/doctrine/jel/doddict/.

[FeH06]   B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, "Protocol Independent Multicast – Sparse Mode (PIM-SM): Protocol Specification (Revised)," RFC 4601, IETF Network Working Group, 2006.

[GAO04]   United States Government Accountability Office, *Defense Acquisitions: the Global Information Grid and challenges facing its implementation,* GAO-04-858, (Washington, DC: July 28, 2004). Available: http://www.gao.gov/new.items/d04858.pdf

[Glo07]   GlobalSecurity.org (2007), "Unmanned Aerial Vehicles," Retrieved May 5, 2007 from http://www.globalsecurity.org/intell/systems/uav-intro.htm.

[Gru07]   S. Gruber, E-mail interview. 30 Oct 2007.

[HoI04]   M. P. Howarth, S. Iyengar, Z. Sun, and H. Cruickshank, "Dynamics of key management in secure satellite multicast," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 2, pp. 308-319, 2004.

[Hub06]   V. Hubenko, "Secure and efficient communications for global information grid users via cooperating space assets," PhD Prospectus, Dept. of Elect. and Comp. Eng., Air Force Institute of Technology, Wright-Patterson AFB, OH, 2006.

[HuR06a] V. Hubenko Jr., R. Raines, M. Temple, R. Mills, and M. Saeger, "Adaptation, Modeling, and Analysis of PIM-DM in a LEO Satellite Network Environment," *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, 2006.

[HuR06b] V. Hubenko Jr., R. Raines, R. Mills, R. Baldwin, B. Mullins and M. Grimaila, "Improving the Global Information Grid's Performance Through Satellite Communications Layer Enhancements," *IEEE Communications,* vol. 44, no. 11, pp. 66-72, 2006.

[HuR07]   V. Hubenko Jr., R. Raines, R. Baldwin, B. Mullins, R. Mills, and M. Grimaila "Improving Satellite Multicast Security Scalability by Reducing Re-keying Requirements," *IEEE Network*, vol. 21, no. 4, pp. 51 -56, 2007.

[HyM07]   M. T. Hyland, B. E. Mullins, R. O. Baldwin, and M. A. Temple, "Simulation-based Performance Evaluation of Mobile Ad Hoc Routing Protocols in a Swarm of Unmanned Aerial Vehicles," *IEEE International Symposium on Pervasive Computing and Ad Hoc Communications (PCAC-07)*, Niagara Falls, Canada, May 2007.

[JiC01]   L. Ji and M. Corson, "Differential Destination Multicast – A MANET Multicast Routing Protocol for Small Groups," *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and communications Societies (INFOCOM)*, pp. 1192 – 1202, Anchorage, AK, April 2001.

[Jor06]   G. Jordt, "Evaluation of Energy Costs and Error Performance of Range-Aware, Anchor-Free Localization Algorithms for Wireless Sensor Networks," Masters Thesis, Dept. of Elect. and Comp. Eng., Air Force Institute of Technology, Wright-Patterson AFB, OH, 2006.

[JuA02]   P. Judge and M. Ammar, "Gothic: a group access control architecture for secure multicast and anycast," *Proceedings of the IEEE INFOCOM*, New York City, NY, USA, 2002.

[JuA03]   P. Judge and M. Ammar, "Security issues and solutions in multicast content distribution: a survey," *IEEE Network*, vol. 17, no. 1, pp. 30-36, 2003.

[KeJ06]   D. Kearney and M. Jasuinas, "Managing power amongst a group of networked embedded FPGAs using dynamic reconfiguration and task migration," *Dagstuhl Seminar Proceedings 06141*, 2006.

[Kru98]   P. Kruus, "A Survey of Multicast Security Issues and Architectures," *Proceedings of the 21st National Information Systems Security Conference*, 1998.

[KuR05]   J.Kurose and K. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet* (3rd Edition). Boston: Pearson Education, 2005.

[LaP06]   L. Lazos and R. Poovendran, "Power proximity based key management for secure multicast in ad hoc networks," *Wireless Networks*, vol. 13, pp. 127 – 148, 2006.

[MaG99]   E. Madruga and J. Garcia-Luna-Aceves, "The Core-Assisted Mesh Protocol," *IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks*, vol. 17, no. 8, pp. 1380 - 1394, Aug 1999.

[Mer07]   Meriam-Webster Online Dictionary (2007), "Autonomy," Retrieved May 18, 2007 from http://www.m-w.com/dictionary/autonomy.

[Mir01]   N. Mir, "A Survey of Data Multicast Techniques, Architectures, and Algorithms," *IEEE Communications Magazine*, pp. 164-170, 2001.

[Mit97]   S. Mittra, "Iolus: A Framework for Scalable Secure Multicasting," *Proc. ACM SIGCOMM '97*, Cannes, France, 1997.

[Moy94]   J.Moy, "Multicast Extensions to OSPF," RFC 1584, IETF Network Working Group, 1994.

[Nat04]   National Institute of Standards and Technology, "Autonomy Levels for Unmanned Systems (ALFUS) Framework," *NIST Special Publication 1011*, version 1.1, 2004. Available: http://www.isd.mel.nist.gov/documents/huang/ALFUS_FrameworkUpdate.pdf

[OSD04]   Office of the Secretary of Defense (OSD), "Defense Science Board Study on Unmanned Aerial Vehicles and Uninhabited Combat Aerial Vehicles," February 2004. Available: http://www.acq.osd.mil/dsb/reports/uav.pdf.

[OSD05]   Office of the Secretary of Defense (OSD), "Unmanned Aircraft Systems Roadmap, 2005–2030," August 2005. Available: http://www.acq.osd.mil/usd/Roadmap%20Final2.pdf.

[OSD07]   Office of the Secretary of Defense (OSD), "Unmanned Systems Roadmap 2007–2032," December 2007. Available: http://www.fas.org/irp/program/collect/usroadmap2007.pdf.

[Pac07]   D. Pack, E-mail interview. 16 May 2007.

[PaO06]   M. Park, N. Okazaki, and S. Seno, "A proposal and its evaluations of a re-keying system for dynamic secure group communications," *Systems and Computers in Japan*, vol. 37, no. 2, pp. 11-24, 2006.

[RaH03]   S. Rafaeli and D. Hutchison, "A survey of key management for secure group communication," *ACM Computing Surveys*, vol. 35, no. 3, pp. 309-329, 2003.

[RoM05]   L. Robertson and T. Meink, "Transformational Communications", Jun 2005. Available: http://www.afrlhorizons.com/Briefs/Jun05/VS0410.html.

[RoP00]   E. Royer and C. Perkins, "Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing", Internet Draft, IETF Mobile Ad Hoc Networking Working Group, 2000.

[SaM00]   L. H. Sahasrabuddhe and B. Mukherjee, "Multicast routing algorithms and protocols: a tutorial," *IEEE Network*, vol. 14, no. 1, pp. 90-102, 2000.

[SiS99]   P. Sinha, R. Sivakumar, and V. Bharghavan, "MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing," IEEE Wireless Communication and Networking Conference, pp. 1313-1317, Sep 1999.

[Spi03]   Spirent Communications, Inc. (2003), "Multicast Routing PIM Sparse Mode and Other Protocols," Retrieved May 13, 2007 from http://www.spirentcom.com/documents/1318.pdf.

[SuH03]   Z. Sun, M. P. Howarth, S. Iyengar, and L. Claverotte, "Networking Issues in IP Multicast over Satellite," *International Journal of Satellite Communications and Networking*, vol. 21, no. pp. 489-507, 2003.

[Ubi07]   Ubiquiti Networks, Inc. (2007) "SuperRange9 datasheet," Retrieved Oct 30, 2007 from http://ubnt.com/downloads/sr9datasheet.pdf.

[USA05a] United States Air Force, *The U.S. Air Force Remotely Piloted Aircraft and Unmanned Aerial Vehicle Strategic Vision*, 2005. Available: http://www.af.mil/shared/media/document/AFD-060322-009.pdf.

[USA05b] United States Air Force Fact Sheets (2005) "Global Hawk," Retrieved May 18, 2007 from http://www.af.mil/factsheets/factsheet.asp?fsID=175.

[Var02]   U. Varshney, "Multicast Over Wireless Networks," *Communications of the ACM*, vol. 45, no. 12, pp. 31-37, 2002.

[WaP98]   D. Waitzman, C. Partridge, and S. Deering, "Distance Vector Multicast Routing Protocol (DVMRP)," RFC 1075, IETF Network Working Group, 1998.

[Wan07]   M. Wancowicz. E-mail interview. 15 May 2007.

[Wik07a] Wikipedia (2007), "Multicast," Retrieved May 5, 2007 from http://en.wikipedia.org/wiki/Multicasting.

[Wik07b] Wikipedia (2007), "MBone," Retrieved May 5, 2007 from http://en.wikipedia.org/wiki/Multicast_backbone.

[WoD07] D. Woods, (2001), "Tutorial: The Wizardry of Multicast," Retrieved Apr 20, 2007 from http://www.networkcomputing.com/1204/1204f1c1.html.

[WuT98] C. Wu, Y. Tay, and C. Toh, "Ad hoc Multicast Routing protocol utilizing Increasing id-numbers (AMRIS) Functional Specification," Internet Draft, IETF MANET Working Group, 1998.

[YaB06] V. Yadav and S.N. Balakrishnan, " Communication control in multiple UAV applications," *AIAA Guidance, Navigation, and Control Conference,* Keystone, CO, United States, vol. 3, pp. 1428-1461, 2006.

[YiL02]   Y. Yi, S. Lee, and W. Su, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," Internet Draft, IETF MANET Working Group, 2002.

Vita

Captain Adrian N. Phillips grew up in Penn Yan, New York and graduated from Penn Yan Academy in 1999. She entered undergraduate studies at Clarkson University in Potsdam, NY where she graduated with a Bachelor of Science degree in Computer Engineering in May 2003. At this time she was also commissioned through the Detachment 536 AFROTC.

Her first assignment was at Wright-Patterson AFB as a Computer Engineer at the National Air and Space Intelligence Center. In July 2005, she became a Deputy Branch Chief. In June 2006, she graduated from Wright State University with a Master's degree in Business Administration. In August 2006, she entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, she will be assigned to the Information Operations Center at Lackland AFB in San Antonio, TX.

REPORT DOCUMENTATION PAGE		*Form Approved* *OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 27-03-2008	2. REPORT TYPE **Master's Thesis**	3. DATES COVERED *(From – To)* August 2006 – March 2008

4. TITLE AND SUBTITLE A Secure Group Communication Architecture for a Swarm of Autonomous Unmanned Aerial Vehicles	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
5. AUTHOR(S) Phillips, Adrian N., Captain, USAF	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GE/ENG/08-09

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) United States Air Force Academy, UAV Research Group Contacts: Daniel Pack, Ph.D. Mailing Address: USAFA/DFEC, 2354 Fairchild Drive, Suite 2F6 USAF Academy, Colorado Springs, CO 80840 Email: Daniel.Pack@usafa.edu Phone: (719) 333-6967 DSN: 333-6967 Fax: (719) 333-3756	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
        APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
    This thesis investigates the application of a secure group communication architecture to a swarm of autonomous unmanned aerial vehicles (UAVs). A multicast secure group communication architecture for the low earth orbit (LEO) satellite environment is evaluated to determine if it can be effectively adapted to a swarm of UAVs and provide secure, scalable, and efficient communications.
    The performance of the proposed security architecture is evaluated with two other commonly used architectures using a discrete event computer simulation developed using MatLab. Performance is evaluated in terms of the scalability and efficiency of the group key distribution and management scheme when the swarm size, swarm mobility, multicast group join and departure rates are varied. The metrics include the total keys distributed over the simulation period, the average number of times an individual UAV must rekey, the average bandwidth used to rekey the swarm, and the average percentage of battery consumed by a UAV to rekey over the simulation period.
    The proposed security architecture can successfully be applied to a swarm of autonomous UAVs using current technology. The proposed architecture is more efficient and scalable than the other tested and commonly-used architectures. Over all the tested configurations, the proposed architecture distributes 55.2 – 94.8% fewer keys, rekeys 59.0 - 94.9% less often per UAV, uses 55.2 - 87.9% less bandwidth to rekey, and reduces the battery consumption by 16.9 – 85.4%.

15. SUBJECT TERMS
   group communication, group key management, multicast, security, swarm, unmanned aerial vehicles

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Barry E. Mullins, Ph.D. (ENG)
REPORT U	ABSTRACT U	c. THIS PAGE U	UU	128	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-6565, ext 3636; e-mail: Barry.Mullins@afit.edu